

PATENT APPLICATION

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re application of

Docket No: Q79956

Christophe PREGUICA, et al.

Appln. No.: 10/787,145

Group Art Unit: 2144

Confirmation No.: 4599

Examiner: Shirley X. ZHANG

Filed: February 27, 2004

For: ADDRESS SEQUENCING IN A DOMAIN NAME SERVER

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

MAIL STOP APPEAL BRIEF - PATENTS

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 41.37, Appellant submits the following:

Table of Contents

I. REAL PARTY IN INTEREST.....	2
II. RELATED APPEALS AND INTERFERENCES	3
III. STATUS OF CLAIMS	4
IV. STATUS OF AMENDMENTS.....	5
V. SUMMARY OF THE CLAIMED SUBJECT MATTER	6
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL	9
VII. ARGUMENT	10
CLAIMS APPENDIX	23
EVIDENCE APPENDIX:	25
RELATED PROCEEDINGS APPENDIX.....	92

I. REAL PARTY IN INTEREST

The real party in interest is ALCATEL. The Assignment is recorded on February 27, 2004 in the U.S. Patent and Trademark Office at Reel 015034, Frame 0305. It is noted that the name of the assignee is now ALCATEL LUCENT.

II. RELATED APPEALS AND INTERFERENCES

Appellant, as well as Appellant's assigns and legal representatives, are unaware of any appeals or interferences which will be directly affected by, or which directly affect or have a bearing on, the Board's decision in the pending case.

III. STATUS OF CLAIMS

Claims 1-4 are all the claims pending in the application, have been finally rejected, and are the subject of this appeal.

Appealed claims are set forth the in Claims Appendix.

IV. STATUS OF AMENDMENTS

No claim amendments were made subsequent to the Final Office Action dated June 10, 2008.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

A summary of the invention is set forth below with reference to non-limiting exemplary embodiments.

The invention is directed to resolution of addresses in a domain name server (DNS) in a telecommunication network. As illustrated in FIG. 1, when a first network element R_1 wants to address a data stream to a second network element R_4 in a communication network, the first element may send to a DNS a request R containing a domain name of the second network element and an IP address of the first element (page 1, lines 12-18, page 5, lines 18-21). The DNS resolves the request and returns to the first element a response R' including an IP address corresponding to the second element. If the domain name of the second element is associated with a plurality of addresses (e.g., a global address and site local address; different types of IPv6 addresses), the DNS sequences the plurality of addresses of the second element and puts the addresses in the order of sequences in a response (page 5, lines 22-24). For instance, if the second element (R_4) has a global address and a site-local address, the DNS determines the type of address to be used and sequences the addresses according to information (e.g., source address, topology) contained in the request received from the first element R_1 , and puts the addresses in the order of sequence in the response R' (page 5, line 25-page 6, line 9). On receipt of the response R' from the DNS, the first element (R_1) determines the address to use and selects the address as a destination address of the second element (page 5, line 18-page 6-line 13). The

DNS server can be implemented in a heterogeneous network composed of IPv4 and IPv 6 network elements (FIG. 2).

Independent Claim 1

1. A domain name server for a data network utilizing the IPv6 protocol stack, said domain name server including:

means for receiving requests adapted to receive a request containing an IPv6 address of a first network element and a domain name; (Fig. 1, D, R; page 5, lines 20-21; Fig. 2, D_B; page 7, line 21-page 8, line 6)

means for returning to the sender of the said request a response containing one or more addresses associated with a second network element corresponding the said domain name; and (Fig. 1, D, R₁, R', R₄; page 5, line 22-page 6, line 9; Fig. 2, X, Y, a_{v6}, a_{v6to4}; page 7, lines 21-25, page 8, lines 7-16, page 9, lines 4-5)

address sequencing means, for sequencing, as a function of said IPv6 address of the first network element, a plurality of IPv6 addresses associated with said second network element, and for putting one or more IPv6 addresses associated with said second network element in the order of the sequence in said response (Fig. 1, D, a_g, a_s; page 4, lines 10-14, page 5, lines 13-14 and 22-24, page 6, lines 7-25; Fig. 2, D_B; page 8, line 17-page 9, line 3).

Dependent Claim 2

2. The domain name server according to claim 1, wherein said address sequencing means is adapted to effect the sequencing as a function of the topology of the network, so that if the IPv6 address of the first network element is a local address belonging to an addressing space and the plurality of addresses associated with the second network element include at least one global IPv6 address and one local IPv6 address belonging to the same addressing space, the more local IPv6 address associated with the second network element corresponding to said addressing space is inserted at a first position within said response. (Fig. 1, page 6, line 22-page 7, line 3)

Dependent Claim 3

3. The domain name server according to claim 1, said address sequencing means is adapted to effect the sequencing so that if the IPv6 address of the first network element is a "6 to 4" type address beginning with the prefix "2002" and the plurality of addresses associated with the second network element include at least one "6 to 4" type address beginning with the prefix "2002", a "6 to 4" type address beginning with the prefix "2002" is inserted at a first position within said response. (Fig. 2, Y, a_{6to4}; page 8, lines 11-16, page 8, line 20-page 9, line 3)

Dependent Claim 4

4. The domain name server according to claim 1, wherein the address sequencing means is adapted to put the sequenced plurality of IPv6 addresses associated with said second network element in said response. (a_g, a_s; page 5, lines 13-14)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Claim 1 is rejected under 35 U.S.C. § 112, 2nd paragraph as having insufficient antecedent basis.

B. Claims 1-4 are rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over IETF Draft by Draves titled "Default Address Selection for IPv6" (hereinafter "Draves"), in view of U.S. Patent No. 6,748,434 to Kavanagh, and the e-mail message posted by Keith Moore on the IETF IPv6 Operations (v6ops) Working Group's discussion board on November 18, 2002 (hereinafter "Moore").

VII. ARGUMENT

A. Claim Rejection under 35 U.S.C. § 112, 2nd paragraph

The Examiner rejected claim 1 on the ground that “the sender” in the recitation of “means for returning to the sender” has insufficient antecedent basis. Appellant, however, respectfully traverses the rejection.

Claim 1 recites, *inter alia*:

“means for receiving requests adapted to receive a request containing an IPv6 address of a first network element and a domain name;
means for returning to the sender of the said request a response containing one or more addresses associated with a second network element corresponding the said domain name;”(emphasis added)

As described above, claim 1 recites “means for receiving requests adapted to receive a request” as a first element and then “means for returning to the sender of the said request a response.” When reading of the term “sender” with regard to the first element, it is evident that the “sender” indicates an entity sending the request to the means for receiving. In addition, since the received request should be sent from a sender, a sender sending the request to the means for receiving is inherently. A received signal always has a sender. In this connection, MPEP § 2173.05(e) (Lack of Antecedent Basis) states:

Obviously, however, the failure to provide explicit antecedent basis for terms does not always render a claim indefinite. If the scope of a claim would be reasonably ascertainable by those skilled in the art, then the claim is not indefinite. >Energizer Holdings Inc. v. Int'l Trade Comm'n, 435 F.3d 1366, 77 USPQ2d 1625 (Fed. Cir. 2006)(holding that "anode gel" provided by implication the antecedent basis for "zinc anode");< Ex parte Porter, 25 USPQ2d 1144, 1145 (Bd. Pat. App. & Inter. 1992) ("controlled stream of fluid" provided reasonable antecedent basis for "the controlled fluid"). Inherent components of elements recited have antecedent basis in the recitation of

the components themselves. For example, the limitation "the outer surface of said sphere" would not require an antecedent recitation that the sphere has an outer surface. See *Bose Corp. v. JBL, Inc.*, 274 F.3d 1354, 1359, 61 USPQ2d 1216, 1218-19 (Fed. Cir 2001) (holding that recitation of "an ellipse" provided antecedent basis for "an ellipse having a major diameter" because "[t]here can be no dispute that mathematically an inherent characteristic of an ellipse is a major diameter"). (emphasis added)

Thus, although the sender does not have an explicit antecedent basis, it does not render the claim to be indefinite because one of ordinary skill in the art could reasonably ascertain the scope regarding the sender, and the inherency of the sender.

Accordingly, Appellant respectfully requests that the rejection be reversed because the description of "the sender" complies with the requirements of 35 U.S.C. § 112, 2nd paragraph.

**B. Claim Rejections under 35 U.S.C. § 103(a) based on Draves in view of
Kavanagh and Moore**

The Examiner rejected claims 1-4 under 35 U.S.C. 103(a) as allegedly being unpatentable over IETF Draft by Draves titled "Default Address Selection for IPv6" (hereinafter "Draves") in view of U.S. Patent No. 6,748,434 to Kavanagh and the e-mail message posted by Keith Moore on the IETF IPv6 Operations (v6ops) Working Group's discussion board on November 18, 2002 (hereinafter "Moore"). Appellant respectfully traverses the rejection.

1. Rejection of Claim 1 based on Draves in view of Kavanagh and Moore

Claim 1 is directed to a domain name server for a data network utilizing the IPv6 protocol stack and recites, *inter alia*:

address sequencing means, for sequencing, as a function of said IPv6 address of the first network element, a plurality of IPv6 addresses associated with said second network element, and for putting one or more IPv6 addresses associated with said second network element in the order of the sequence in said response.

a. Draves does not teach or suggest the “address sequencing means” recited in claim 1

Draves teaches default address selection algorithms for IPv6. More specifically, Draves teaches a source address selection algorithm and a destination address algorithm for multiple addresses (sections 5 and 6; pages 8-12).

In the rejection of claim 1, the Examiner asserted that Draves (section 6 “Destination Address Section”, Rules 4, 5, 7) teaches “the address sequencing means, for sequencing, as a function of said IPv6 address of the first network element, a plurality of IPv6 address associated with said second network element,” as recited in claim 1 (page 5 of the Final Office Action). In addition, the Examiner acknowledged that Draves does not expressly teach “the address sequence sequencing is performed on the DNS server and the DNS server then puts one or more IPv6 addresses associated with said second network element in the order of the sequence in said response” (page 5 of the Final Office Action).

On page 2 of the Advisory Action, however, the Examiner altered his position on Draves and contended that Draves expressly teaches “putting IPv6 addresses associated with said second network element in the order of the sequence in said response.” Regarding the DNS server, the Examiner stated that Draves does not expressly teach that the addressing sequencing is performed on the DNS server (page 5 of the Final Office Action; page 2 of the Advisory Action).

Appellant submits that, contrary to the Examiner's assertion, Draves does not teach or suggest the claimed address sequencing means as will now be discussed below.

Firstly, as to the DNS server, the Examiner stated that Draves does not expressly teach that the addressing sequencing is performed on the DNS server. The Examiner is partially correct. Draves does not teach that the sequencing of a plurality of IPv6 addresses associated with the second network element is performed by a DNS. The Examiner does not, however, acknowledge or recognize that Draves explicitly states that the sorting of the destination addresses is performed by a node, not the DNS server. With respect to the destination addresses selection, Draves states at the Introduction (paragraph 4, page 3) as follows:

In the case of destination address selection, the DNS may return a set of addresses for a given name, and an application needs to decide which one to use first, and in what order to try others should the first one not be reachable. The destination address ordering rules in section 6, when applied to the set of addresses returned by the DNS, provide such a recommended ordering. (emphasis added)

The Draves description teaches that a node queries a DNS to resolve a name and then the DNS returns to the node a set of addresses comprising e.g., a global IPv6 address and a global IPv4 address (Introduction; paragraph 3, page 2). The node then uses a destination address selection algorithm for choosing an address among the set of addresses. More specifically, the node's destination address selection algorithm takes a list of destination addresses and sorts the addresses to produce a new list according to specified rules (section 6, lines 1-2, Rules 1-10). Accordingly, Draves explicitly teaches the sorting of the destination addresses is performed by a node rather than suggesting the DNS server performing the addressing sequencing required in claim 1.

In addition, contrary to the Examiner's assertion, Draves does not teach or suggest "the address sequencing means for putting one or more IPv6 addresses associated with said second network element in the order of the sequence in said response", as recited in claim 1.

In Draves, the destination address selection algorithm sorts a list of destination addresses to produce a new list by applying pair-wise comparison rules 1-10 in order (section 6, paragraph 5; page 11). Thus, the resultant list includes ordered addresses which the node uses to connect to another device in turn. This sorting of the addresses in the list, however, does not correspond to the claimed putting one or more IPv6 addresses in the order of sequence in the response which is returned from the DNS to the sender.

The reason that the Draves' sorting of addresses cannot be interpreted as the claimed putting the IPv6 addresses in a response is because Draves destination algorithm merely decides and produces the ordering of the use of addresses received from the DNS, but has nothing to do with producing a response including the addresses to be sent to a node.

Accordingly, Draves does not teach or suggest the DNS server putting the addresses in the response sent to the sender of the request as recited in claim 1.

b. Kavanagh does not make up for the deficiencies of Draves with respect to claim 1

The Examiner alleged that Kavanagh compensates for Draves' missing elements in support of the rejection of claim 1. In particular, on page 5 of the Final Office Action, the

Examiner contended that the description on the domain name server (DNS) teaches the missing elements of Draves (Fig. 2, col. 2, lines 1-27; col. 7, lines 45-49; col.9, lines 9-19).

Kavanagh teaches a method for providing adaptive node selection in a network using a DNS server which monitors the status of network elements (Fig. 2).

Kavanagh, however, fails to teach or suggest “a data network utilizing the IPv6 protocol stack,” as recited in claim 1. As described in the present application (pages 1 and 2) and Draves (section 1), a IPV 6 addressing architecture allows a network element to have multiple IP addresses assigned. This causes problems of a source address selection and a destination address selection specific to IPv6 networks. Kavanagh, however, does not teach IPv6 networks.

Further, Kavanagh does not teach or suggest “sequencing a plurality of IPv6 addresses associated with said second network element,” as recited in claim 1. Although Kavanagh describes “a plurality of addresses” that the DNS returns to a node, the addresses are not associated with “a specific one network element” in a network. Rather, the addresses represent a plurality of nodes in the network. In Kavanagh, a node selector filters network nodes and/or network links (col. 2, lines 1-11). The nodes and links are represented by IP addresses (col. 2, line 9). Hence, the entities subjected to selecting, grouping and/or preference ordering are respective nodes represented by respective addresses. In addition, the criteria taught by Kavanagh to effect the grouping or filtering are nodal criteria such as geographical location, functionality, or capacity (col. 2, lines 25-27), i.e. criteria that refer to features of the respective nodes (see also “local GGSN”; col. 9, lines 29-36). This fact about the nodes’ addresses is clearly taught from the Kavanagh description that states:

The SGSN1 sends the DNS query to the DNS server in the GPRS backbone looking for the APN symbolic name "my.isp.net" to be resolved. The DNS server returns the list of IP addresses (i.e., GGSNs) for the APN. The order of the list returned to the SGSN1 is typically random (e.g., GGSN1, GGSN3, GGSN2). (col. 4, line 66-col. 5, line 3) (emphasis added)

As a result, the adaptive node selector has the option to return multiple IP addresses, namely the IP addresses of GGSN2 and GGSN3 for the APN requested (i.e., "my.isp.net"). However, the adaptive node selector notices that the request for this APN came from SGSN2 in the South Site and consequently returns the IP address of GGSN3 to SGSN2 and filters out the IP address of GGSN2. (col. 9, lines 9-15) (emphasis added)

As such, Kavanagh teaches a plurality of addresses for different network elements, i.e. GGSNs (Figs. 1 and 3), but fails to teach or suggest the claimed plurality of IPv6 addresses associated with said second network element recited in claim 1.

Therefore, Kavanagh fails to teach or suggest the address sequencing means for sequencing a plurality of IPv6 addresses associated with a given network element, as required in claim 1. Accordingly, Appellant respectfully submits that Kavanagh does not make up for the deficiencies of Draves.

c. One of ordinary skill in the art would not have been motivated to combine Draves and Kavanagh

The Examiner alleged that Moore provides motivation to one of ordinary skill in the art to combine Draves and Kavanagh. In support of the rejection of claim 1, the Examiner cited to Moore's email message and the e-messages prior to Moor's email, asserting that Moore's disclosure is clearly directed to a DNS server and is sufficient to motivate one of ordinary skill

in the art to implement Draves' address selection algorithm in Kavanagh's domain name server (page 6 of the Final Office Action).

In the Advisory Action, the Examiner contended that Moore does not have to teach a DNS server, but merely needs to suggest using a server to choose address in order to motivate one of ordinary skill in the art to combine Draves and Kavanagh. The Examiner also alleged that the disclosure of "getaddrinfo address ordering" and the statement of "I believe what you need us some (dynamic) server selection method" in Moore would have led one of ordinary skill in the art to conclude Moore's suggesting a DNS server (page 2 of the Advisory Action).

The Examiner, however, did not provide any support on how description of a server choosing an address would sufficiently motivate one of ordinary skill in the art to combine Draves and Kavanagh to arrive at the claimed domain name server.

Further, contrary to the Examiner's assertion, one of ordinary skill in the art would not conclude Moore's disclosure as suggesting a DNS server. Also, the Examiner incorrectly recognized Moore's disclosure and this incorrect recognition resulted from impermissible hindsight reasoning by relying on Appellant's disclosure to find the claimed invention obvious. The Moore's email, "Re: getaddrinfo address ordering", reads as follows.

I believe what is needed is to stop relying so much on applications/hosts choosing destination addresses. Hosts should have as few addresses as possible. The network should make a best effort to deliver the traffic to whatever address is used over the links permitted for such use. (emphasis added)

The socket `getaddrinfo()` is a generic function through which an application requests an operating system to convert a name into an address. It should be noted that the `getaddrinfo()` is not at all specific to DNS servers and is implemented by a variety of network elements, including any IP host, through a variety of lower layer functions, e.g. NIS, `/etc/hosts`, WINS, Netbios.

Moore merely states that the number of addresses returned by the network should be reduced. Nothing in the above statement addresses the DNS server sequencing, “as a function of [an] address of the first network element”, a plurality of addresses of a second network element and “putting [plurality of addresses] in the order of the sequence in [a] response, as required in claim 1. Rather, Moore simply indicates that the reliance on the host (i.e. a node) should be reduced by reducing the number of addresses from which the nodes had to choose. It is evident that Moore was still suggesting that the application/hosts choose destination addresses. (see. “stop relying so much”)

Appellant now turns to the whole thread of e-mails from which Moore’s disclosure is extracted in order to clarify the technical content of the disclosure in view of the context in which it appeared. (see <http://www.ops.ietf.org/lists/v6ops/v6ops.2002/maillist.html#00887>)

Looking at the date index of the thread, the discussion “Re: `getaddrinfo` address ordering” starts with message number #00856, dated 27 Nov 2002 09:22:04 +0200 and ends with message #00887, dated 28 Nov 2002 08:01:39 –0500.

In the message #00856, the discussion starts on the topic of enhancing a `getaddrinfo` resolver in an IP host (*we would want to add some extra smarts to `getaddrinfo`*). The resolver is called by an application (*you need the address ordering to be sensitive to both network*

configuration and to the particular application.). The resolver processes DNS records returned by a DNS server with some default ordering. The getaddrinfo resolver at the IP host is intended to return the records to the calling application with a potentially different order (*you want to influence the default DNS record ordering*). Thus, the discussion initially focuses on an IP host, e.g. a DNS client, and an application run thereon. Besides, this initial focus remains the same until message #00883. Also, it is noteworthy that message #00860 explicitly refers to Draves.

In Message #00867, Moore gives his own vision of how the network could help in the process of address ordering. The local network would provide the IP host with a configuration file that specifies the preferred ordering (*eventually it might be nice if the host could get that preference list from the local network*). It should be also noted that messages #00878 and #00883 focus on the application run by the IP host.

Then, for the first time, in YOSHIFUJI's message #00886, the focus is shifted from the IP host to "some server." YOSHIFUJI's statement, however, is very elliptical. No information is provided about the undefined server. This statement starts to broaden the focus of the discussion, but does not provide any clear direction. It is worth noting that no one in the recipients elaborated on YOSHIFUJI's idea. This may indicate that YOSHIFUJI's statement was not even understood.

Moore's response in message #00887 does not elaborate on "some server selection method". Quite oppositely, it broadens the discussion one step further by referring to "the network" as a whole, which implies that the focus proposed by YOSHIFUJI's was either not understood or not considered relevant by Moore. Moore's message #00867 already specified the

way the network could contribute to address selection. By referring to the network, message #00887 appears to hint at that earlier post by the same author.

Besides, Moore's message #00887 further broadens the topic of the discussion from address selection to delivery of traffic (*to deliver the traffic to whatever address is used over the links permitted for such use.*). Hence, it presents a broad perspective in a way that typically ends a discussion without deciding any practical conclusion or action point. Indeed no one responded.

Further, YOSHIFUJI's statement is not enabling and fails to point to any specific server whereas possibilities are innumerable (e.g. a server dedicated to selection services). Thus, one cannot reasonably assert that YOSHIFUJI was inherently referring to a DNS server.

As such, in view of the context in which Moore was written, Moore fails to teach or suggest that a DNS server needs to be modified to sequence a plurality of addresses. Accordingly, Appellant respectfully submits that Moore's disclosure does not provide any motivation to combine Draves and Kavanagh to produce the claimed invention.

d. A supposed combination of Draves and Kavanagh would not produce the invention of claim 1

As discussed above, Draves and Kavanagh, alone or in combination, fail to teach or suggest all the elements of claim 1. Accordingly, Appellant submits that a supposed combination of Draves and Kavanagh would not produce the invention of claim 1 contrary to the Examiner's assertion.

For at least reasons above, Appellant respectfully submits that claim 1 would not have been obvious under 35 U.S.C. § 103(a) over Draves in view of Kavanagh and Moore and thus the rejection of claim 1 should be reversed.

2. Rejection of Claims 2-4 based on Draves in view of Kavanagh and Moore

The Examiner rejected dependent claims 2-4 as being unpatentable over Draves in view of Kavanagh and Moore. Claims 2-4, however, should be patentable over the references at least because of their dependencies from claim 1.

In rejection of **claim 2**, the Examiner asserted that Draves (section 6. Destination Address Selection Rule 8) teaches the claimed recitation. As discussed with respect to claim 1, the references fail to teach or suggest all the elements of claim 1 on which claim 2 depends. Accordingly, Appellant submits that claim 2 is patentable over the references at least because of its dependency on claim 1. In addition, since Draves does not teaches sequencing and putting addresses in a response from a DNS to a sender, the Rule 8 of the Destination Address Selection does not correspond to the recitation of claim 2.

With respect to **claim 3**, the Examiner contended that Rule 5 (Prefer matching label) of Draves teaches the claimed recitation. Since the references fail to teach or suggest all the elements of claim 1, claim 3 is patentable by virtue of its dependency on claim 1.

Regarding **claim 4**, the Examiner alleged that Draves in view of Kavanagh teaches the claimed elements and Moore provides motivation to combine Draves and Kavanagh. As discussed with respect to claim 1, however, Draves and Kavanagh do not teach or suggest all the

APPEAL BRIEF UNDER 37 C.F.R. § 41.37
U.S. Appln. No.: 10/787,145

Atty. Docket No.: Q79956

elements of claim 1 and Moore would not have motivated one of ordinary skill in the art to combine Draves and Kavanagh. Accordingly, claim 4 is patentable over the references at least because of its dependency from claim 1.

The USPTO is directed and authorized to charge the statutory fee (37 C.F.R. §41.37(a) and 1.17(c)) and all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,



Kelly G. Hyndman
Registration No. 39,234

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE

23373

CUSTOMER NUMBER

Date: April 9, 2009

CLAIMS APPENDIX

CLAIMS 1-4 ON APPEAL:

1. A domain name server for a data network utilizing the IPv6 protocol stack, said domain name server including:

means for receiving requests adapted to receive a request containing an IPv6 address of a first network element and a domain name;

means for returning to the sender of the said request a response containing one or more addresses associated with a second network element corresponding the said domain name; and

address sequencing means, for sequencing, as a function of said IPv6 address of the first network element, a plurality of IPv6 addresses associated with said second network element, and for putting one or more IPv6 addresses associated with said second network element in the order of the sequence in said response.

2. The domain name server according to claim 1, wherein said address sequencing means is adapted to effect the sequencing as a function of the topology of the network, so that if the IPv6 address of the first network element is a local address belonging to an addressing space and the plurality of addresses associated with the second network element include at least one global IPv6 address and one local IPv6 address belonging to the same addressing space, the more local IPv6 address associated with the second network element corresponding to said addressing space is inserted at a first position within said response.

3. The domain name server according to claim 1, said address sequencing means is adapted to effect the sequencing so that if the IPv6 address of the first network element is a "6 to 4" type address beginning with the prefix "2002" and the plurality of addresses associated with the second network element include at least one "6 to 4" type address beginning with the prefix "2002", a "6 to 4" type address beginning with the prefix "2002" is inserted at a first position within said response.

4. The domain name server according to claim 1, wherein the address sequencing means is adapted to put the sequenced plurality of IPv6 addresses associated with said second network element in said response .

EVIDENCE APPENDIX:

Submitted herewith are printouts of email messages (#00856-#00887) that appear at the website (<http://www.ops.ietf.org/lists/v6ops/v6ops.2002/maillist.html>) discussed in the Pre-Appeal Brief dated December 10, 2009 and this Appeal Brief. The printouts are a whole thread of emails posted on the IETF IPv6 Operations (v6ops) Working Group's discussion board from which Keith Moore's email (#00887) was extracted. One of Keith Moore's email messages was relied upon by the Examiner as to the grounds of rejection to be reviewed on appeal (page 6 of the Non-Final Office Action dated November 26, 2007; page 4 of the Final Office Action dated June 10, 2008; page 2 of the Advisory Action dated September 30, 2008). This evidence should be admitted because the copies are a printout version of the email messages including Moore's email that the Examiner relied on and are specified in the Pre-Appeal Brief. Further, the whole thread of email messages is necessary to clarify the technical content of the Moore's email in the context of all the messages exchanged under the same topic. The single email fails to provide a complete picture. The whole message thread was reviewed in detail after receiving the Advisory Action. Accordingly, the evidence should be considered as entered in the record or be admitted pursuant to 37 C.F.R. § 41.37(c)(1)(ix).

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

getaddrinfo address ordering [Re: IPv6 transition architectediscussion]

- *To:* Keith Moore <moore@cs.utk.edu>
 - *Subject:* getaddrinfo address ordering [Re: IPv6 transition architectediscussion]
 - *From:* Pekka Savola <pekkas@netcore.fi>
 - *Date:* Wed, 27 Nov 2002 09:22:04 +0200 (EET)
 - *Cc:* Joshua Graessley <jgraessley@apple.com>, <v6ops@ops.ietf.org>
 - *Delivery-date:* Tue, 26 Nov 2002 23:23:55 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* <200211252110.gAPLA9112170@astro.cs.utk.edu>
 - *Sender:* owner-v6ops@ops.ietf.org
-

On Mon, 25 Nov 2002, Keith Moore wrote:

[...]

> at least not right now. attempts to talk to v6 sites typically
> time out (which is annoying because the v6 addresses are tried first -
> another thing which needs to be fixed)

Do I sense this as a voice for support to modify or at least explore the current de-facto standard ordering?

> > We were considering turning on 6to4 by
> > default when there are no routing advertisements. This working group
> > has dissuaded us from doing so. If we did turn on 6to4, we would want
> > to add some extra smarts to getaddrinfo to list IPv4 addresses first
> > when 6to4 is in use to reduce the load on the relays.
>
> it turns out that you need the address ordering to be sensitive to both
> network configuration and to the particular application. some apps will
> be v4 by default, others will be v6 default, others will be exclusively
> one or the other.

There are already some variables that affect this, namely:

- 1) whether the DNS lookup produces A, AAAA, or both; ie "remote site configuration"
- 2) what ordering for DNS records is used when both are used (default is AAAA first everywhere I know, no possibility to change that)
- 3) whether the application uses AF_INET, AF_INET6 or AF_UNSPEC (with server apps, this is even a bit muddier.)

These help quite a bit, but I guess adding some getaddrinfo hint like AI_PREFERV4 or AI_PREFERV6 could be added in the case that DNS lookup returns both addresses, you use AF_UNSPEC and you want to influence the

default DNS record ordering.

--

Pekka Savola "Tell me of difficulties surmounted,
Netcore Oy not those you stumble over and fall"
Systems. Networks. Security. -- Robert Jordan: A Crown of Swords

- **Follow-Ups:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Craig Metz <cmetz@inner.net>
- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Keith Moore <moore@cs.utk.edu>
- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Gert Doering <gert@space.net>
- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* itojun@iijlab.net

- **References:**

- **Re: IPv6 transition architecture discussion**
 - *From:* Keith Moore <moore@cs.utk.edu>

- Prev by Date: **Re: 6to4 security questions**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: IPv6 transition architecture discussion**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* Pekka Savola <pekkas@netcore.fi>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* itojun@iijlab.net
 - *Date:* Wed, 27 Nov 2002 16:27:18 +0900
 - *Cc:* v6ops@ops.ietf.org
 - *Delivery-date:* Tue, 26 Nov 2002 23:27:28 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* pekkas's message of Wed, 27 Nov 2002 09:22:04 +0200. <Pine.LNX.4.44.0211270915100.28193-100000@netcore.fi>
 - *Sender:* owner-v6ops@ops.ietf.org
-

>These help quite a bit, but I guess adding some getaddrinfo hint like
>AI_PREFERV4 or AI_PREFERV6 could be added in the case that DNS lookup
>returns both addresses, you use AF_UNSPEC and you want to influence the
>default DNS record ordering.

i don't think it wise to have flag bit - if you pass AI_PREFERV4
you will stuck with IPv4 even when IPv6 provides much better
connectivity, and the curse won't go away until you recompile.

leave it to default address selection rule and logic in getaddrinfo(3).

itojun

-
- **Follow-Ups:**
 - **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>
 - **References:**
 - **getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>
 - Prev by Date: **getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Previous by thread: **getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Index(es):

- **Date**
- **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]

- *To:* itojun@iijlab.net
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]
 - *From:* Pekka Savola <pekkas@netcore.fi>
 - *Date:* Wed, 27 Nov 2002 09:32:51 +0200 (EET)
 - *Cc:* v6ops@ops.ietf.org
 - *Delivery-date:* Tue, 26 Nov 2002 23:33:39 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* <20021127072718.907C84B22@coconut.itojun.org>
 - *Sender:* owner-v6ops@ops.ietf.org
-

On Wed, 27 Nov 2002 itojun@iijlab.net wrote:

```
> >These help quite a bit, but I guess adding some getaddrinfo hint like
> >AI_PREFERV4 or AI_PREFERV6 could be added in the case that DNS lookup
> >returns both addresses, you use AF_UNSPEC and you want to influence the
> >default DNS record ordering.
>
> i don't think it wise to have flag bit - if you pass AI_PREFERV4
> you will stuck with IPv4 even when IPv6 provides much better
> connectivity, and the curse won't go away until you recompile.
```

That's still better than apps shipping with AF_INET or no getaddrinfo at all, right?

```
> leave it to default address selection rule and logic in getaddrinfo(3).
```

I'm not sure how default address selection, at least ones I've seen would help with this.

Perhaps they don't implement `_destination_` address selection.

--

Pekka Savola	"Tell me of difficulties surmounted,
Netcore Oy	not those you stumble over and fall"
Systems. Networks. Security.	-- Robert Jordan: A Crown of Swords

-
- **Follow-Ups:**
 - **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* itojun@iijlab.net

- **References:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* itojun@iijlab.net

- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* Pekka Savola <pekkas@netcore.fi>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* itojun@iijlab.net
 - *Date:* Wed, 27 Nov 2002 16:38:04 +0900
 - *Cc:* v6ops@ops.ietf.org
 - *Delivery-date:* Tue, 26 Nov 2002 23:38:20 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* pekkas's message of Wed, 27 Nov 2002 09:32:51 +0200. <Pine.LNX.4.44.0211270929540.28616-100000@netcore.fi>
 - *Sender:* owner-v6ops@ops.ietf.org
-

```
>>      leave it to default address selection rule and logic in getaddrinfo(3).
>I'm not sure how default address selection, at least ones I've seen would
>help with this.
>Perhaps they don't implement _destination_ address selection.
```

they do. see draft-ietf-ipngwg-default-addr-select-06.txt section 5.

itojun

- **Follow-Ups:**
 - **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* JINMEI Tatuya / □\$B?@L@C#:H□(B <jinmei@isl.rdc.toshiba.co.jp>
- **References:**
 - **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>
- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]

- *To:* v6ops@ops.ietf.org
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]
 - *From:* JINMEI Tatuya / □\$B?@L@C#:H□(B <jimmei@isl.rdc.toshiba.co.jp>
 - *Date:* Wed, 27 Nov 2002 17:16:39 +0900
 - *Delivery-date:* Wed, 27 Nov 2002 00:17:25 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* <20021127073804.0F2094B24@coconut.itojun.org>
 - *Organization:* Research & Development Center, Toshiba Corp., Kawasaki, Japan.
 - *References:* <20021127073804.0F2094B24@coconut.itojun.org>
 - *Sender:* owner-v6ops@ops.ietf.org
 - *User-agent:* Wanderlust/2.6.1 (Upside Down) Emacs/21.2 Mule/5.0 (SAKAKI)
-

>>>> On Wed, 27 Nov 2002 16:38:04 +0900,
 >>>> itojun@iijlab.net said:

>>> leave it to default address selection rule and logic in getaddrinfo(3).
 >> I'm not sure how default address selection, at least ones I've seen would
 >> help with this.
 >> Perhaps they don't implement `_destination_` address selection.

> they do. see draft-ietf-ipngwg-default-addr-select-06.txt section 5.

Section 10.3 of draft-ietf-ipv6-default-addr-select-09.txt (this is the latest draft I believe) may also help. KAME snaps have getaddrinfo considering the policy table and a tool to modify the policy table. I guess Windows XP support the feature, too.

Whether we should change the default is, of course, another issue.

JINMEI, Tatuya
 Communication Platform Lab.
 Corporate R&D Center, Toshiba Corp.
jimmei@isl.rdc.toshiba.co.jp

- **References:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* itojun@iijlab.net

- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* Pekka Savola <pekkas@netcore.fi>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* Gert Doering <gert@space.net>
 - *Date:* Wed, 27 Nov 2002 09:50:43 +0100
 - *Cc:* Keith Moore <moore@cs.utk.edu>, Joshua Graessley <jgraessley@apple.com>, v6ops@ops.ietf.org
 - *Delivery-date:* Wed, 27 Nov 2002 00:51:58 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* <Pine.LNX.4.44.0211270915100.28193-100000@netcore.fi>; from pekkas@netcore.fi on Wed, Nov 27, 2002 at 09:22:04AM +0200
 - *References:* <200211252110.gAPLA9l12170@astro.cs.utk.edu> <Pine.LNX.4.44.0211270915100.28193-100000@netcore.fi>
 - *Sender:* owner-v6ops@ops.ietf.org
 - *User-agent:* Mutt/1.2.5.1i
-

Hi,

On Wed, Nov 27, 2002 at 09:22:04AM +0200, Pekka Savola wrote:

> These help quite a bit, but I guess adding some getaddrinfo hint like
> AI_PREFERV4 or AI_PREFERV6 could be added in the case that DNS lookup
> returns both addresses, you use AF_UNSPEC and you want to influence the
> default DNS record ordering.

In addition to that, provide a system configuration item that will set the default preference in case the application doesn't specify AI_PREFER*

Usually it's not something the application would care about, but more the system administrator - "I want to have IPv6 but I do not trust it enough to be the default" vs. "I know IPv6 is now working well enough".

Gert Doering
-- NetMaster

--
Total number of prefixes smaller than registry allocations: 50279 (49875)

SpaceNet AG	Mail: netmaster@Space.Net
Joseph-Dollinger-Bogen 14	Tel : +49-89-32356-0
80807 Muenchen	Fax : +49-89-32356-299

- **Follow-Ups:**
 - **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>
- **References:**
 - **Re: IPv6 transition architecture discussion**
 - *From:* Keith Moore <moore@cs.utk.edu>
 - **getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>
- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering [Re: IPv6 transition architectediscussion]

- *To:* Gert Doering <gert@space.net>
- *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architectediscussion]
- *From:* Pekka Savola <pekkas@netcore.fi>
- *Date:* Wed, 27 Nov 2002 12:16:03 +0200 (EET)
- *Cc:* Keith Moore <moore@cs.utk.edu>, Joshua Graessley <jgraessley@apple.com>, <v6ops@ops.ietf.org>
- *Delivery-date:* Wed, 27 Nov 2002 02:17:42 -0800
- *Envelope-to:* v6ops-data@psg.com
- *In-reply-to:* <20021127095043.Q15927@Space.Net>
- *Sender:* owner-v6ops@ops.ietf.org

On Wed, 27 Nov 2002, Gert Doering wrote:

```
> On Wed, Nov 27, 2002 at 09:22:04AM +0200, Pekka Savola wrote:
> > These help quite a bit, but I guess adding some getaddrinfo hint like
> > AI_PREFERV4 or AI_PREFERV6 could be added in the case that DNS lookup
> > returns both addresses, you use AF_UNSPEC and you want to influence the
> > default DNS record ordering.
>
> In addition to that, provide a system configuration item that will
> set the default preference in case the application doesn't specify
> AI_PREFER*
```

Yes, that could be the policy table (or my suggestion) actually implementing the toggle in the resolver, often in /etc/resolv.conf.

```
> Usually it's not something the application would care about, but more
> the system administrator - "I want to have IPv6 but I do not trust it
> enough to be the default" vs. "I know IPv6 is now working well enough".
```

Totally agree -- some apps can be exceptions though.

I want to be able to start enabling v6 by default, to build the user base, but I don't want it to cause any ill effects to apps or the user yet.

Currently there is no way to do this, and I believe it's critical when we really want to deploy v6 to the masses.

--

Pekka Savola "Tell me of difficulties surmounted,
Netcore Oy not those you stumble over and fall"
Systems. Networks. Security. -- Robert Jordan: A Crown of Swords

-
- **Follow-Ups:**
 - **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* "Jeroen Massar" <jeroen@unfix.org>
 - **References:**
 - **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Gert Doering <gert@space.net>
 - Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Next by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Next by thread: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* "Pekka Savola" <pekkas@netcore.fi>, "Gert Doering" <gert@space.net>
 - *Subject:* RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* "Jeroen Massar" <jeroen@unfix.org>
 - *Date:* Wed, 27 Nov 2002 13:06:22 +0100
 - *Cc:* "Keith Moore" <moore@cs.utk.edu>, "Joshua Graessley" <jgraessley@apple.com>, <v6ops@ops.ietf.org>
 - *Delivery-date:* Wed, 27 Nov 2002 04:07:07 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *Importance:* Normal
 - *In-reply-to:* <Pine.LNX.4.44.0211271214020.29842-100000@netcore.fi>
 - *Organization:* Unfix
 - *Sender:* owner-v6ops@ops.ietf.org
-

Pekka Savola wrote:

```
> On Wed, 27 Nov 2002, Gert Doering wrote:
> > On Wed, Nov 27, 2002 at 09:22:04AM +0200, Pekka Savola wrote:
> > > These help quite a bit, but I guess adding some
> > getaddrinfo hint like
> > > AI_PREFERV4 or AI_PREFERV6 could be added in the case
> > that DNS lookup
> > > returns both addresses, you use AF_UNSPEC and you want to
> > influence the
> > > default DNS record ordering.
> >
> > In addition to that, provide a system configuration item that will
> > set the default preference in case the application doesn't specify
> > AI_PREFER*
>
> Yes, that could be the policy table (or my suggestion) actually
> implementing the toggle in the resolver, often in /etc/resolv.conf.
>
> > Usually it's not something the application would care
> > about, but more
> > the system administrator - "I want to have IPv6 but I do
> > not trust it
> > enough to be the default" vs. "I know IPv6 is now working
> > well enough".
>
> Totally agree -- some apps can be exceptions though.
>
> I want to be able to start enabling v6 by default, to build
> the user base,
> but I don't want it to cause _any_ ill effects to apps or the
> user yet.
```

>
> Currently there is no way to do this, and I believe it's
> critical when we really want to deploy v6 to the masses.

The OS or actually the resolver part of the OS should be able to be configured with an option exactly like above. Programs should never have a preference, if they do they should ask for an AF_INET or AF_INET6 that's enough preference already :)

I am against having the AI_PREFER4/6 as this would be too program specific and one needs to recompile to get it out.

The resolver could/should have a setting in f.e /etc/resolv.conf or in the registry: "prefer AAAA,A" This would first try IPv6, then IPv4. I think I once saw a discussion on the debian-ipv6 lists talking about such a mechanism which would be implemented into the glibc resolver library.

One could also have a tool which periodically, say once a week, or on-boot checks if there is a working IPv6 uplink. Eg by sending a ping packet to a well known host by doing this in IPv4 and IPv6 this toggle could be flipped automatically. Ofcourse due to privacy there should be an easy way to turn this off and to change the host.

Greetings,
Jeroen

• **Follow-Ups:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Stig Venaas <Stig.Venaas@uninett.no>

• **References:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>

- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* Jeroen Massar <jeroen@unfix.org>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* Stig Venaas <Stig.Venaas@uninett.no>
 - *Date:* Wed, 27 Nov 2002 13:28:02 +0100
 - *Cc:* "Pekka Savola" <pekkas@netcore.fi>, "Gert Doering" <gert@space.net>, "Keith Moore" <moore@cs.utk.edu>, "Joshua Graessley" <jgraessley@apple.com>, v6ops@ops.ietf.org
 - *Delivery-date:* Wed, 27 Nov 2002 04:29:04 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* <[000801c2960d\\$676308c0\\$210d640a@unfix.org](mailto:000801c2960d$676308c0$210d640a@unfix.org)>; from jeroen@unfix.org on Wed, Nov 27, 2002 at 01:06:22PM +0100
 - *References:* <Pine.LNX.4.44.0211271214020.29842-100000@netcore.fi> <[000801c2960d\\$676308c0\\$210d640a@unfix.org](mailto:000801c2960d$676308c0$210d640a@unfix.org)>
 - *Sender:* owner-v6ops@ops.ietf.org
 - *User-agent:* Mutt/1.2.5.1i
-

On Wed, Nov 27, 2002 at 01:06:22PM +0100, Jeroen Massar wrote:
 > The OS or actually the resolver part of the OS should be able
 > to be configured with an option exactly like above.

Yes, that is a good idea.

> Programs should never have a preference, if they do they
 > should ask for an AF_INET or AF_INET6 that's enough preference already
 > :)
 > I am against having the AI_PREFER4/6 as this would be too program
 > specific and one needs to recompile to get it out.

I agree, we really should avoid this. I think adding even more complexity to getaddrinfo() is a bad idea. And not only does it take time to deploy, but it might also be hard to change the applications later. This is really something the sysadmin should be able to choose, there is no default that fits all.

Stig

- **Follow-Ups:**
 - **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>

- **References:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>
- **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* "Jeroen Massar" <jeroen@unfix.org>

- Prev by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* Pekka Savola <pekkas@netcore.fi>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* Keith Moore <moore@cs.utk.edu>
 - *Date:* Wed, 27 Nov 2002 08:39:13 -0500
 - *Cc:* Keith Moore <moore@cs.utk.edu>, Joshua Graessley <jgraessley@apple.com>, v6ops@ops.ietf.org
 - *Delivery-date:* Wed, 27 Nov 2002 05:43:17 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* (Your message of "Wed, 27 Nov 2002 09:22:04 +0200.") <Pine.LNX.4.44.0211270915100.28193-100000@netcore.fi>
 - *Sender:* owner-v6ops@ops.ietf.org
-

> > at least not right now. attempts to talk to v6 sites typically
> > time out (which is annoying because the v6 addresses are tried first -
> > another thing which needs to be fixed)
>
> Do I sense this as a voice for support to modify or at least explore the
> current de-facto standard ordering?

I'm certainly going to change it on the machines that I use
(it helps that I have source code)

> > it turns out that you need the address ordering to be sensitive to both
> > network configuration and to the particular application. some apps will
> > be v4 by default, others will be v6 default, others will be exclusively
> > one or the other.

>
> There are already some variables that affect this, namely:
>
> 1) whether the DNS lookup produces A, AAAA, or both; ie "remote
> site configuration"

I don't see how this affects ordering - if you don't get back A
records are you going to put the AAAA records in a different
order relative to one another than if you do get them back?

> 2) what ordering for DNS records is used when both are used
> (default is AAAA first everywhere I know, no possibility to change that)
>
> 3) whether the application uses AF_INET, AF_INET6 or AF_UNSPEC
> (with server apps, this is even a bit muddier.)
>
> These help quite a bit, but I guess adding some getaddrinfo hint like
> AI_PREFERV4 or AI_PREFERV6 could be added in the case that DNS lookup
> returns both addresses, you use AF_UNSPEC and you want to influence the

> default DNS record ordering.

it's fairly easy for an app to reorder addresses that it gets back between IPv4 and IPv6. but I often find myself wanting to specify a mixed preference like:

6to4 addresses
native IPv4
other IPv6

Keith

- **Follow-Ups:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>

- **References:**

- **getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>

- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

Re: getaddrinfo address ordering [Re: IPv6 transition architectediscussion]

- *To:* Keith Moore <moore@cs.utk.edu>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architectediscussion]
 - *From:* Pekka Savola <pekkas@netcore.fi>
 - *Date:* Wed, 27 Nov 2002 15:45:25 +0200 (EET)
 - *Cc:* Joshua Graessley <jgraessley@apple.com>, <v6ops@ops.ietf.org>
 - *Delivery-date:* Wed, 27 Nov 2002 05:45:41 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* <200211271339.gARDdDI28938@astro.cs.utk.edu>
 - *Sender:* owner-v6ops@ops.ietf.org
-

On Wed, 27 Nov 2002, Keith Moore wrote:

```
> > There are already some variables that affect this, namely:
> >
> > 1) whether the DNS lookup produces A, AAAA, or both; ie "remote
> > site configuration"
>
> I don't see how this affects ordering - if you don't get back A
> records are you going to put the AAAA records in a different
> order relative to one another than if you do get them back?
```

Of course you're right -- the ordering described here is only relevant when both are used.

The point was that, because of current ordering, people don't really put v6 addresses in AAAA fields that much, so this is no problem (address resolves to either A or AAAA (under e.g. ipv6 subdomain)).

But we'd like it to be different..

```
> > 2) what ordering for DNS records is used when both are used
> > (default is AAAA first everywhere I know, no possibility to change that)
> >
> > 3) whether the application uses AF_INET, AF_INET6 or AF_UNSPEC
> > (with server apps, this is even a bit muddier.)
> >
> > These help quite a bit, but I guess adding some getaddrinfo hint like
> > AI_PREFERV4 or AI_PREFERV6 could be added in the case that DNS lookup
> > returns both addresses, you use AF_UNSPEC and you want to influence the
> > default DNS record ordering.
>
> it's fairly easy for an app to reorder addresses that it gets back
> between IPv4 and IPv6. but I often find myself wanting to specify
> a mixed preference like:
>
```

> 6to4 addresses

.. particularly if you use 6to4 addresses yourself; otherwise you really don't buy much by separating them from other v6.

> native IPv4
> other IPv6

Yeah -- but that's something that we must not try to plug into every app -- it should really be done by lower layers (at least, by functions provided by lower layers). No need to write the code 1000 times :-).

--

Pekka Savola "Tell me of difficulties surmounted,
Netcore Oy not those you stumble over and fall"
Systems. Networks. Security. -- Robert Jordan: A Crown of Swords

• **Follow-Ups:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Keith Moore <moore@cs.utk.edu>

• **References:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Keith Moore <moore@cs.utk.edu>

- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* Pekka Savola <pekkas@netcore.fi>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* Keith Moore <moore@cs.utk.edu>
 - *Date:* Wed, 27 Nov 2002 08:58:32 -0500
 - *Cc:* Keith Moore <moore@cs.utk.edu>, Joshua Graessley <jgraessley@apple.com>, v6ops@ops.ietf.org
 - *Delivery-date:* Wed, 27 Nov 2002 06:01:59 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* (Your message of "Wed, 27 Nov 2002 15:45:25 +0200.") <Pine.LNX.4.44.0211271541290.31686-100000@netcore.fi>
 - *Sender:* owner-v6ops@ops.ietf.org
-

> > it's fairly easy for an app to reorder addresses that it gets back
> > between IPv4 and IPv6. but I often find myself wanting to specify
> > a mixed preference like:
> >
> > 6to4 addresses
>
> .. particularly if you use 6to4 addresses yourself; otherwise you really
> don't buy much by separating them from other v6.
>
> > native IPv4
> > other IPv6
>
> Yeah -- but that's something that we must not try to plug into every app
> -- it should really be done by lower layers (at least, by functions
> provided by lower layers). No need to write the code 1000 times :-).

personally I intend to rewrite the library routine so that it references
a config file that specifies preferred ordering. eventually it might be
nice if the host could get that preference list from the local network.

Keith

-
- **References:**
 - **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>
 - Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**

- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering [Re: IPv6 transition architectediscussion]

- *To:* Stig Venaas <Stig.Venaas@uninett.no>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architectediscussion]
 - *From:* Pekka Savola <pekkas@netcore.fi>
 - *Date:* Wed, 27 Nov 2002 16:53:12 +0200 (EET)
 - *Cc:* Jeroen Massar <jeroen@unfix.org>, "Gert Doering" <gert@space.net>, "Keith Moore" <moore@cs.utk.edu>, "Joshua Graessley" <jgraessley@apple.com>, <v6ops@ops.ietf.org>
 - *Delivery-date:* Wed, 27 Nov 2002 06:54:48 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* <20021127132801.A13186@sverresborg.uninett.no>
 - *Sender:* owner-v6ops@ops.ietf.org
-

On Wed, 27 Nov 2002, Stig Venaas wrote:

```
> > Programs should _never_ have a preference, if they do they
> > should ask for an AF_INET or AF_INET6 that's enough preference already
> > :)
> > I am against having the AI_PREFER4/6 as this would be too program
> > specific and one needs to recompile to get it out.
>
> I agree, we really should avoid this. I think adding even more
> complexity to getaddrinfo() is a bad idea. And not only does it
> take time to deploy, but it might also be hard to change the
> applications later. This is really something the sysadmin should
> be able to choose, there is no default that fits all.
```

There are more layers than that, e.g.

```
- "if I have an app and install it anywhere at all, I have no way of
knowing whether the system has implemented A,AAA preferation method -- so
I'll just ship it with AF_UNSPEC + AI_PREFERV4, so people can run it with
'app v6host', 'app -6 v46host' or 'app v46host' without problems, it'll
always fall back to v4 if it doesn't know better"
```

```
- "my app is special and is really only usable in v6 context; I'd prefer
it to use v6 even though whatever's configured in A,AAAA preferation
method. AF_UNSPEC + AI_PREFERV6 could handle that"
```

Ie. I'd like the app writers to be able to write "v6-safe" code. Code that, when run or compiled on nearly any system, would produce code that's safe to be used in v4-preferring environments _regardless of_ whether the app user's box even has any A,AAAA preferation method.

But I can very well like with stuff like AI_PREFER* -- we just need the toggle to change the ordering of A,AAAA records when using AF_UNSPEC ASAP, if not yesterday. If we could manage to get that deployed fast enough,

and v4 preferred widely enough, perhaps in 1-2 years people could start shipping v6-enabled apps, waiting for the site admins decision to prefer v6 instead when they're confident enough to do so.

--

Pekka Savola "Tell me of difficulties surmounted,
Netcore Oy not those you stumble over and fall"
Systems. Networks. Security. -- Robert Jordan: A Crown of Swords

-
- **Follow-Ups:**
 - **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* "Jeroen Massar" <jeroen@unfix.org>
 - **References:**
 - **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Stig Venaas <Stig.Venaas@uninett.no>
 - Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Next by thread: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* Pekka Savola <pekkas@netcore.fi>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* Craig Metz <cmetz@inner.net>
 - *Date:* Wed, 27 Nov 2002 10:39:57 -0500
 - *Cc:* v6ops@ops.ietf.org
 - *Delivery-date:* Wed, 27 Nov 2002 07:41:08 -0800
 - *Envelope-to:* v6ops-data@psg.com
 - *In-reply-to:* Your message of "Wed, 27 Nov 2002 09:22:04 +0200." <Pine.LNX.4.44.0211270915100.28193-100000@netcore.fi>
 - *Sender:* owner-v6ops@ops.ietf.org
-

In message <Pine.LNX.4.44.0211270915100.28193-100000@netcore.fi>, you write:

>Do I sense this as a voice for support to modify or at least explore the
>current de-facto standard ordering?

There is no specified standard, deliberately. This is an implementation issue. Not all implementations have this problem, but many people on this list don't seem to understand that and instead blame `getaddrinfo()` or blame IPv6.

>These help quite a bit, but I guess adding some `getaddrinfo` hint like
>`AI_PREFERV4` or `AI_PREFERV6` could be added in the case that DNS lookup
>returns both addresses, you use `AF_UNSPEC` and you want to influence the
>default DNS record ordering.

NO! NO! NO!

`getaddrinfo` is a PROTOCOL INDEPENDENT API. It is NEVER appropriate to cruft ANY protocol-specific options into `getaddrinfo`. Protocol-specific hacks belong in protocol-specific APIs. If the protocol-independent APIs do not give you the features you want, then use the protocol-dependent APIs instead, which will give you more specific controls.

If an application wants to have `getaddrinfo()` return things in a particular order, it should call `getaddrinfo()` multiple times in the order it wants, i.e., call it first for `req.ai_addr=AF_INET`, second for `req.ai_addr=AF_INET6`, etc. HOWEVER, this behavior would be TOTALLY BROKEN as it first defeats protocol independence (the whole point; if you're not using `getaddrinfo` in a protocol independent way, why are you using it at all?), and second because it limits you to IPv4 and IPv6.

BSDI did this right. BSD/OS had a configurable resolver return order, I believe through a file and an environment variable. Didn't like the system default? Users can request things be done in a different order. Problem solved.

I think BSDI gave their code to ISC for use in BIND, but it appears that this feature got lost in the transition. Still, if someone with clues asked the old BSDI folks nicely, you might be able to get their code for this and fold it into your favorite implementation.

-Craig

- **Follow-Ups:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Keith Moore <moore@cs.utk.edu>
- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>

- **References:**

- **getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>

- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]

- *To:* Craig Metz <cmetz@inner.net>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]
 - *From:* Pekka Savola <pekkas@netcore.fi>
 - *Date:* Wed, 27 Nov 2002 18:12:41 +0200 (EET)
 - *Cc:* v6ops@ops.ietf.org
 - *In-reply-to:* <200211271432.gAREWOnF009847@inner.net>
-

On Wed, 27 Nov 2002, Craig Metz wrote:

```
> In message <Pine.LNX.4.44.0211270915100.28193-100000@netcore.fi>, you write:
> >Do I sense this as a voice for support to modify or at least explore the
> >current de-facto standard ordering?
>
> There is no specified standard, deliberately. This is an implementation
> issue. Not all implementations have this problem, but many people on this list
> don't seem to understand that and instead blame getaddrinfo() or blame IPv6.
```

Yes, this is implementation specific. But most do have this `_operational_`
`_deployment_` `_problem_`. And sitting on our thumbs does little to help
with getting v6 out there. I'd like to see it one day :-)

```
> >These help quite a bit, but I guess adding some getaddrinfo hint like
> >AI_PREFERV4 or AI_PREFERV6 could be added in the case that DNS lookup
> >returns both addresses, you use AF_UNSPEC and you want to influence the
> >default DNS record ordering.
>
> NO! NO! NO!
>
> getaddrinfo is a PROTOCOL INDEPENDENT API. It is NEVER appropriate to cruft
> ANY protocol-specific options into getaddrinfo. Protocol-specific hacks belong
> in protocol-specific APIs. If the protocol-independent APIs do not give you
> the features you want, then use the protocol-dependent APIs instead, which
> will give you more specific controls.
```

I'm sorry to wake you up from the wonderland, but getaddrinfo has already
been encumbered with protocol-specific options (think `AI_MAPPED..`).

Whether anything can or should be salvaged is another issue..

```
> If an application wants to have getaddrinfo() return things in a particular
> order, it should call getaddrinfo() multiple times in the order it wants, i.e.,
> call it first for req.ai_addr=AF_INET, second for req.ai_addr=AF_INET6, etc.
> HOWEVER, this behavior would be TOTALLY BROKEN as it first defeats protocol
> independence (the whole point; if you're not using getaddrinfo in a protocol
> independent way, why are you using it at all?), and second because it limits
> you to IPv4 and IPv6.
```

Right. (I prefer some deliberate ordering for AF_UNSPEC of course.)

> BSDI did this right. BSD/OS had a configurable resolver return order, I
> believe through a file and an environment variable. Didn't like the system
> default? Users can request things be done in a different order. Problem solved.

That's nice, but as more folks haven't done it -- it helps little in this operational problem.

This is probably caused by the fact that most many not have considered this problem serious at all, worth spending time on -- and there are many strategies how v6 could be deployed, some of which aren't affected by the lack of the option. I believe it's our job to raise awareness on this.

--

Pekka Savola "Tell me of difficulties surmounted,
Netcore Oy not those you stumble over and fall"
Systems. Networks. Security. -- Robert Jordan: A Crown of Swords

- **Follow-Ups:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Craig Metz <cmetz@inner.net>

- **References:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Craig Metz <cmetz@inner.net>

- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* "Pekka Savola" <pekkas@netcore.fi>, "Stig Venaas" <Stig.Venaas@uninett.no>
- *Subject:* RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
- *From:* "Jeroen Massar" <jeroen@unfix.org>
- *Date:* Wed, 27 Nov 2002 17:21:53 +0100
- *Cc:* "Gert Doering" <gert@space.net>, "Keith Moore" <moore@cs.utk.edu>, "Joshua Graessley" <jgraessley@apple.com>, <v6ops@ops.ietf.org>
- *In-reply-to:* <Pine.LNX.4.44.0211271645350.32265-100000@netcore.fi>
- *Organization:* Unfix

Pekka Savola [<mailto:pekkas@netcore.fi>] wrote:

```
> On Wed, 27 Nov 2002, Stig Venaas wrote:
> > > Programs should _never_ have a preference, if they do they
> > > should ask for an AF_INET or AF_INET6 that's enough
> preference already
> > > :)
> > > I am against having the AI_PREFER4/6 as this would be too program
> > > specific and one needs to recompile to get it out.
> >
> > I agree, we really should avoid this. I think adding even more
> > complexity to getaddrinfo() is a bad idea. And not only does it
> > take time to deploy, but it might also be hard to change the
> > applications later. This is really something the sysadmin should
> > be able to choose, there is no default that fits all.
>
> There are more layers than that, e.g.
>
> - "if I have an app and install it anywhere at all, I have no way of
> knowing whether the system has implemented A,AAA preferation
> method -- so
> I'll just ship it with AF_UNSPEC + AI_PREFERV4, so people can
> run it with
> 'app v6host', 'app -6 v46host' or 'app v46host' without
> problems, it'll
> always fall back to v4 if it doesn't know better"
```

It will currently always fall back to v4 if one wrote it correctly:

- tries AAAA (if dns returned it)
- tries A (if dns returned it)
- > - "my app is special and is really only usable in v6
- > context; I'd prefer
- > it to use v6 even though whatever's configured in A,AAA preferation
- > method. AF_UNSPEC + AI_PREFERV6 could handle that"

One should use AF_INET6 for that. And this is what currently happens. But indeed if the resolver reorders A's in front of AAAA this doesn't work anymore.

Then again the application can request a getaddrinfo() for AF_INET6 and if those didn't connect go for the AF_INET addresses.

I don't think that there are many, even better name me 1, application that would want to do it. Even though it could be a quite plausible example one day. I don't think it will happen.

> Ie. I'd like the app writers to be able to write "v6-safe"
 > code. Code
 > that, when run or compiled on nearly any system, would
 > produce code that's
 > safe to be used in v4-preferring environments _regardless of_
 > whether the
 > app user's box even has any A,AAAA preferation method.

Ofcourse this should be true. And currently it nicely falls back to v4. if the resolver doesn't support preferring they should push their resolver coder to upgrade it. If people don't upgrade reguraly they won't have IPv6 support anyways on most platforms

> But I can very well like with stuff like AI_PREFER* -- we
 > just need the toggle to change the ordering of A,AAAA records when using
 > AF_UNSPEC ASAP, if not yesterday.
 > If we could manage to get that deployed fast enough,
 > and v4 preferred widely enough, perhaps in 1-2 years people
 > could start shipping v6-enabled apps, waiting for the site admins
 > decision to prefer v6 instead when they're confident enough to do so.

IMHO the toggle should go into the resolver and NOT in the application. If the application is specific enough to always request eg IPv6 then it should do getaddrinfo(AF_INET6) and then fallback to getaddrinfo(AF_INET).

As been said getaddrinfo is for AF independence if you depend on something request specifically that type of protocol.

Greets,
 Jeroen

• **Follow-Ups:**

- **RE: getaddrinfo address ordering [Re: IPv6 transition architectediscussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>

• **References:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architectediscussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>

• **Prev by Date: Re: getaddrinfo address ordering [Re: IPv6 transition architectediscussion]**

- Next by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

RE: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]

- *To:* Jeroen Massar <jeroen@unfix.org>
- *Subject:* RE: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]
- *From:* Pekka Savola <pekkas@netcore.fi>
- *Date:* Wed, 27 Nov 2002 18:31:46 +0200 (EET)
- *Cc:* "Stig Venaas" <Stig.Venaas@uninett.no>, "Gert Doering" <gert@space.net>, "Keith Moore" <moore@cs.utk.edu>, "Joshua Graessley" <jgraessley@apple.com>, <v6ops@ops.ietf.org>
- *In-reply-to:* <000901c29631\$1a6166b0\$210d640a@unfix.org>

On Wed, 27 Nov 2002, Jeroen Massar wrote:

```
> > On Wed, 27 Nov 2002, Stig Venaas wrote:
> > > Programs should _never_ have a preference, if they do they
> > > should ask for an AF_INET or AF_INET6 that's enough
> > preference already
> > > :)
> > > I am against having the AI_PREFER4/6 as this would be too program
> > > specific and one needs to recompile to get it out.
> > >
> > > I agree, we really should avoid this. I think adding even more
> > > complexity to getaddrinfo() is a bad idea. And not only does it
> > > take time to deploy, but it might also be hard to change the
> > > applications later. This is really something the sysadmin should
> > > be able to choose, there is no default that fits all.
> >
> > There are more layers than that, e.g.
> >
> > - "if I have an app and install it anywhere at all, I have no way of
> > knowing whether the system has implemented A,AAA preferation
> > method -- so
> > I'll just ship it with AF_UNSPEC + AI_PREFERV4, so people can
> > run it with
> > 'app v6host', 'app -6 v46host' or 'app v46host' without
> > problems, it'll
> > always fall back to v4 if it doesn't know better"
>
> It will currently always fall back to v4 if one wrote it correctly:
> - tries AAAA (if dns returned it)
> - tries A (if dns returned it)
```

Sure.. but in this case www.google.com will not have AAAA records in, well, about 5-10 years. They have no incentive to do so, at all -- quite the contrary -- as their v4-enabled users would get worse service.

Or many more such sites.

If try A first would be the option, www.google.com (etc.) folks would have _no_ reason not to put the AAAA record in www.google.com in a year or two.

But this is just v6 deployment strategy. Both ways could work.. but it seems to me that we must go and take 1).

```
> > - "my app is special and is really only usable in v6
> > context; I'd prefer
> > it to use v6 even though whatever's configured in A,AAAA preferation
> > method. AF_UNSPEC + AI_PREFERV6 could handle that"
>
> One should use AF_INET6 for that. And this is what currently happens.
> But indeed if the resolver reorders A's in front of AAAA this doesn't
> work anymore.
> Then again the application can request a getaddrinfo() for AF_INET6 and
> if those didn't connect go for the AF_INET addresses.
> I don't think that there are many, even better name me 1, application
> that would want to do it. Even though it could be a quite plausible
> example one day. I don't think it will happen.
```

This is a bit dubious case, I agree -- but e.g. future peer-to-peer, end-to-end applications could profit from trying v6 first, then falling back to v4 if necessary.

```
> > Ie. I'd like the app writers to be able to write "v6-safe"
> > code. Code
> > that, when run or compiled on nearly any system, would
> > produce code that's
> > safe to be used in v4-preferring environments _regardless of_
> > whether the
> > app user's box even has any A,AAAA preferation method.
>
> Ofcourse this should be true. And currently it nicely falls back
> to v4.
```

You're missing the point.

```
> if the resolver doesn't support preferring they should
> push their resolver coder to upgrade it.
```

A lot of pushing is needed, it seems.

```
> > But I can very well like with stuff like AI_PREFER* -- we
> > just need the toggle to change the ordering of A,AAAA records when
> using
> > AF_UNSPEC ASAP, if not yesterday.
> > If we could manage to get that deployed fast enough,
> > and v4 preferred widely enough, perhaps in 1-2 years people
> > could start shipping v6-enabled apps, waiting for the site admins
> > decision to prefer v6 instead when they're confident enough to do so.
>
> IMHO the toggle should go into the resolver and NOT in the application.
> If the application is specific enough to always request eg IPv6 then it
> should do getaddrinfo(AF_INET6) and then fallback to
> getaddrinfo(AF_INET).
```

I kinda agree with this.

--

Pekka Savola

"Tell me of difficulties surmounted,

Netcore Oy not those you stumble over and fall"
Systems. Networks. Security. -- Robert Jordan: A Crown of Swords

- **Follow-Ups:**
 - **Re: getaddrinfo address ordering**
 - *From:* YOSHIFUJI Hideaki / □\$B5HF#1QL@□(B <yoshfuji@wide.ad.jp>
- **References:**
 - **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* "Jeroen Massar" <jeroen@unfix.org>
- Prev by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering**
- Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

RE: 6to4 security questions

- *To:* "Alain Durand" <Alain.Durand@Sun.COM>
 - *Subject:* RE: 6to4 security questions
 - *From:* "Christian Huitema" <huitema@windows.microsoft.com>
 - *Date:* Wed, 27 Nov 2002 09:03:35 -0800
 - *Cc:* "Brian E Carpenter" <brian@hursley.ibm.com>, "Pekka Savola" <pekkas@netcore.fi>, <v6ops@ops.ietf.org>
-

> > There are however a number of mitigating factors:
> >
> > 1) The attack does not include a multiplier effect; the amount of
> > traffic directed at the target will be about equal to the amount of
> > traffic sent by the attacker.
> >
> Not necessarily. It depend on the protocol used on the reflectors.
> If you can find a UDP protocol that echos n packets to an original
> short incoming packet, you have a multiplier effect.

The problem there is with the applications, not with 6to4. Deploying a protocol that is ready to blast packets without some form of initial handshake is a truly bad idea. Let's face it: as an application, you can never trust that the source address is not forged.

> > 2) The attack packets go through a choke point, the 6to4 relay
between
> > the laundering site and the target.
> >
> Not necessarily. If thousands of reflectors are used, they will use
> different relays.

Well, it is still mitigation: it forces the attacker to find thousand of relays, and thousands of applications.

> > 3) The packets received by the target contain the address of the
> > relaying 6to4 site.
> >
> If (many) thousands of reflectors are used, each reflector can be
> exercised only once and still create a massive DDOS.
> Such an attack will be even harder to detect using packet sampling
> techniques.

I am not sure that we cannot use packet sampling techniques, actually. Suppose that each 6to4 router, with a low probability of e.g. 1%, sends an ITRACE or equivalent to the IPv6 source of the encapsulated packet, and that the ITRACE contains the indication of the actual IPv4 source. If thousands of relays are used, tens of them will send an ITRACE packet towards the target of the attack, thus exposing the IPv4 origin. I guess we ought to study this ITRACE idea further.

> >4) The payload of the packets received by the target will be a response
> >generated by the laundering server, which limits any "magic packet"
> >issue.
> >
> If many reflectors are exercised, depending on the protocol used,
> each DDOS packet may be different.

No, that is not true in practice. Even if they are different, they cannot be "chosen". The "magic packet" terminology refers to attacks such as "ping of death", in which a carefully crafted packet triggers an abnormal condition in the receiving host. In a reflection attack, the packet received by the target is a response packet, produced by a normally behaving server. Such packets are likely to be discarded without much processing, since the target did not initiate the transaction and is not expecting a response; they are also very unlikely to be crafted in the specific way that may trigger a software bug. For example, you cannot mount a SYN attack using a reflector: the target will receive a SYN ACK, that will be immediately discarded since there is no corresponding half-open connection.

The consequence is that a reflected attack has to be brute force, aiming simply at the bandwidth of the target. Such attacks require massive amounts of packets, and are thus much more susceptible to detection using ITRACE.

> >5) The attack only provides value if the attacker's IPv4 connection was
> >subject to ingress filtering, which is alas not a very common case.
> >
> True. But this will create disincentive for ISP to put ingress filtering
> in place and ruin the effort of those who did.

Uh? If you are saying that the ISP can somehow guarantee to their customers that all source addresses are genuine, you are seriously misguided. For example, an attacker could hack into a router, from where it can send packets from virtually any source addresses. There are thousands of routers in the Internet; are you willing to assume that none of them will ever be cracked?

> >Because of the absence of a magic packet effect, this attack is only
> >really powerful if it is practiced by a "fleet of zombies" using a large
> >number of reflectors.
> >
> Not really, you can use only a limited number of zombies (even a well
> connected one is enough) if you can discover thousands of valid 6to4
> addresses.
> As the packets will be 'landed' by the 6to4 routers, it will be very
> hard
> to trace the attack back to the zombie.

If you have a limited number of zombies, then the ITRACE approach will work.

> >In short, yes it is a vulnerability, but it is not a terribly dangerous
> >one, and it is a vulnerability that will in any case disappear with
> >6to4, when sites receive native IPv6 connectivity. So, yes, a fix is

> >welcome; however, the fix should not be so drastic as to impede the
> >"autonomous deployment" advantage of 6to4.
> >
> See my comment in the other thread on 6to4 & Teredo vs Tunnel Broker.

I believe that 6to4 and tunnel broker have some very different characteristics. Opposing one to the other as the ultimate solution is probably a bad idea. It is much more reasonable to acknowledge the difference and let the market sort it out.

-- Christian Huitema

-
- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Next by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - Previous by thread: **Re: 6to4 security questions**
 - Next by thread: **ISP Cases Draft: Which sections have too much information?**
 - Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* "Pekka Savola" <pekkas@netcore.fi>, "Jeroen Massar" <jeroen@unfix.org>
 - *Subject:* RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* "Christian Huitema" <huitema@windows.microsoft.com>
 - *Date:* Wed, 27 Nov 2002 09:06:56 -0800
 - *Cc:* "Stig Venaas" <Stig.Venaas@uninett.no>, "Gert Doering" <gert@space.net>, "Keith Moore" <moore@cs.utk.edu>, "Joshua Graessley" <jgraessley@apple.com>, <v6ops@ops.ietf.org>
-

> > It will currently always fall back to v4 if one wrote it correctly:
> > - tries AAAA (if dns returned it)
> > - tries A (if dns returned it)
>
> Sure.. but in this case www.google.com will not have AAAA records in,
> well, about 5-10 years. They have no incentive to do so, at all --
> quite the contrary -- as their v4-enabled users would get worse
service.

A nice point of 6to4 as a transition strategy is that its performance are guaranteed to be very close from those of the underlying IPv4. A normal evolution for sites like Google would be to multihomed and expose both a native IPv6 connection and a 6to4 connection. Address selection rules will guarantee that "transition" users pick the 6to4 destination, while native users pick the native address. This will largely deal with your performance concern.

-- Christian Huitema

- **Follow-Ups:**

- **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>

- Prev by Date: **RE: 6to4 security questions**
- Next by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **SUMMARY: 6to4 security**
- Next by thread: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

RE: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]

- *To:* Christian Huitema <huitema@windows.microsoft.com>
 - *Subject:* RE: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]
 - *From:* Pekka Savola <pekkas@netcore.fi>
 - *Date:* Wed, 27 Nov 2002 19:13:42 +0200 (EET)
 - *Cc:* Jeroen Massar <jeroen@unfix.org>, Stig Venaas <Stig.Venaas@uninett.no>, Gert Doering <gert@space.net>, Keith Moore <moore@cs.utk.edu>, Joshua Graessley <jgraessley@apple.com>, <v6ops@ops.ietf.org>
 - *In-reply-to:* <DAC3FCB50E31C54987CD10797DA511BA1D77BE@WIN-MSG-10.wingroup.windeploy.ntdev.microsoft.com>
-

On Wed, 27 Nov 2002, Christian Huitema wrote:

```
> > > It will currently always fall back to v4 if one wrote it correctly:
> > > - tries AAAA (if dns returned it)
> > > - tries A (if dns returned it)
> >
> > Sure.. but in this case www.google.com will not have AAAA records in,
> > well, about 5-10 years. They have no incentive to do so, _at all_ --
> > quite the contrary -- as their v4-enabled users would get worse
> > service.
>
> A nice point of 6to4 as a transition strategy is that its performance
> are guaranteed to be very close from those of the underlying IPv4. A
> normal evolution for sites like Google would be to multihomed and expose
> both a native IPv6 connection and a 6to4 connection. Address selection
> rules will guarantee that "transition" users pick the 6to4 destination,
> while native users pick the native address. This will largely deal with
> your performance concern.
```

This would help only if all v6 users, v6-native included, would enable 6to4 as an "optimization technique", as proposed previously.

Doesn't seem like a scalable solution.

(why v6-native? v6 native connection, here meaning non-6to4 global addresses, cannot be guaranteed to reach google in an optimal fashion -- quite far from it.)

Another transition strategy would be to _not_ use any native v6 _at all_ yet, only 6to4 -- or feed the hosts only native more specific routes to those networks that are "well-connected" (where you wouldn't use 6to4).

Reminds too much of Jim Flemings ideas, too scary to even contemplate.

--

Pekka Savola "Tell me of difficulties surmounted,
Netcore Oy not those you stumble over and fall"
Systems. Networks. Security. -- Robert Jordan: A Crown of Swords

- **Follow-Ups:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Keith Moore <moore@cs.utk.edu>

- **References:**

- **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* "Christian Huitema" <huitema@windows.microsoft.com>

- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* Craig Metz <cmetz@inner.net>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* Keith Moore <moore@cs.utk.edu>
 - *Date:* Wed, 27 Nov 2002 12:12:02 -0500
 - *Cc:* Pekka Savola <pekkas@netcore.fi>, v6ops@ops.ietf.org
 - *In-reply-to:* (Your message of "Wed, 27 Nov 2002 10:39:57 EST.")
<200211271432.gAREWOnF009847@inner.net>
-

> NO! NO! NO!

>

> getaddrinfo is a PROTOCOL INDEPENDENT API. It is NEVER appropriate to cruft
> ANY protocol-specific options into getaddrinfo. Protocol-specific hacks belong
> in protocol-specific APIs. If the protocol-independent APIs do not give you
> the features you want, then use the protocol-dependent APIs instead, which
> will give you more specific controls.

you're absolutely correct. thank you for saying that.

Keith

- **References:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Craig Metz <cmetz@inner.net>

- Prev by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **RE: IPv6 transition architecture discussion**
- Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- **To:** Pekka Savola <pekkas@netcore.fi>
 - **Subject:** Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - **From:** Keith Moore <moore@cs.utk.edu>
 - **Date:** Wed, 27 Nov 2002 12:23:06 -0500
 - **Cc:** Christian Huitema <huitema@windows.microsoft.com>, Jeroen Massar <jeroen@unfix.org>, Stig Venaas <Stig.Venaas@uninett.no>, Gert Doering <gert@space.net>, Keith Moore <moore@cs.utk.edu>, Joshua Graessley <jgraessley@apple.com>, v6ops@ops.ietf.org
 - **In-reply-to:** (Your message of "Wed, 27 Nov 2002 19:13:42 +0200.") <Pine.LNX.4.44.0211271908540.1843-100000@netcore.fi>
-

> > A nice point of 6to4 as a transition strategy is that its performance
> > are guaranteed to be very close from those of the underlying IPv4. A
> > normal evolution for sites like Google would be to multihomed and expose
> > both a native IPv6 connection and a 6to4 connection. Address selection
> > rules will guarantee that "transition" users pick the 6to4 destination,
> > while native users pick the native address. This will largely deal with
> > your performance concern.
>
> This would help only if all v6 users, v6-native included, would enable
> 6to4 as an "optimization technique", as proposed previously.

I don't buy it because it's an all-or-nothing argument. v6 deployment isn't going to be uniform across the world, or across applications. Some places will deploy native v6 aggressively; other places will feel more satisfied with their existing v4 infrastructures. Some applications will naturally favor v6, others will favor v4, others will be agnostic about it.

Keith

• References:

- **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - **From:** Pekka Savola <pekkas@netcore.fi>
- Prev by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: getaddrinfo address ordering**
- Previous by thread: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**

68

- **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering

- *To:* pekkas@netcore.fi
 - *Subject:* Re: getaddrinfo address ordering
 - *From:* YOSHIFUJI Hideaki / □\$B5HF#1QL@□(B <yoshfuji@wide.ad.jp>
 - *Date:* Thu, 28 Nov 2002 02:37:40 +0900 (JST)
 - *Cc:* jeroen@unfix.org, Stig.Venaas@uninett.no, gert@space.net, moore@cs.utk.edu, jgraessley@apple.com, v6ops@ops.ietf.org
 - *In-reply-to:* <Pine.LNX.4.44.0211271823380.1181-100000@netcore.fi>
 - *References:* <[000901c29631\\$1a6166b0\\$210d640a@unfix.org](mailto:000901c29631$1a6166b0$210d640a@unfix.org)><Pine.LNX.4.44.0211271823380.1181-100000@netcore.fi>
-

[post by non-subscriber. with the massive amount of spam, it is easy to miss and therefore delete mis-posts. so fix subscription addresses!]

In article <Pine.LNX.4.44.0211271823380.1181-100000@netcore.fi> (at Wed, 27 Nov 2002 18:31:46 +0

```
> > It will currently always fall back to v4 if one wrote it correctly:
> > - tries AAAA (if dns returned it)
> > - tries A (if dns returned it)
>
> Sure.. but in this case www.google.com will not have AAAA records in,
> well, about 5-10 years. They have no incentive to do so, _at all_ --
> quite the contrary -- as their v4-enabled users would get worse service.
```

This problem is NOT only ipv6 and/or getaddrinfo thing at all.
If one of IPv4 address on a DNS for a single FQDN were down,
what would be happened?

(You might have recognized similar problem with `h_addr_list[]` in `struct hostent{}` :--)

What should be to blame is poor programming of application.

--

Hideaki YOSHIFUJI @ USAGI Project <yoshfuji@linux-ipv6.org>
GPG FP: 9022 65EB 1ECF 3AD1 0BDF 80D8 4807 F894 E062 0EEA

- **Follow-Ups:**
 - **Re: getaddrinfo address ordering**
 - *From:* Stig Venaas <Stig.Venaas@uninett.no>

- **References:**

- **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* "Jeroen Massar" <jeroen@unfix.org>
- **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>

- Prev by Date: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **Re: ISP Cases Draft: Which sections have too much information?**
- Previous by thread: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]

- *To:* Pekka Savola <pekkas@netcore.fi>
 - *Subject:* Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]
 - *From:* Craig Metz <cmetz@inner.net>
 - *Date:* Wed, 27 Nov 2002 11:46:10 -0500
 - *Cc:* v6ops@ops.ietf.org
 - *In-reply-to:* Your message of "Wed, 27 Nov 2002 18:12:41 +0200." <Pine.LNX.4.44.0211271804020.954-100000@netcore.fi>
-

In message <Pine.LNX.4.44.0211271804020.954-100000@netcore.fi>, you write:
>I'm sorry to wake you up from the wonderland, but getaddrinfo has already
>been encumbered with protocol-specific options (think AI_MAPPED...)
>
>Whether anything can or should be salvaged is another issue..

Last time I checked, AI_MAPPED was not a valid req.ai_flags argument to getaddrinfo(), only to the IPv6-specific API getnodeinfo(). That was a reasonable way to do things -- protocol independent APIs are protocol independent, and protocol-specific features are available through protocol specific APIs.

Admittedly, the last time I checked was a long time ago, so I should check to see whether there has been devolution in this way in the mean time.

>That's nice, but as more folks haven't done it -- it helps little in this
>operational problem.

>
>This is probably caused by the fact that most many not have considered
>this problem serious at all, worth spending time on -- and there are many
>strategies how v6 could be deployed, some of which aren't affected by the
>lack of the option. I believe it's our job to raise awareness on this.

Raising awareness is fine. But that twenty, thirty, forty emails on this mailing list on this topic don't actually cause the changes to happen. A polite email to your implementation's authors ("vendor") pointing out this problem, pointing out how it was solved years ago, and pointing out where the code might be freely available (I don't know that it is for certain, that's a slightly informed speculation), would do far more good towards getting your implementation to fix the problem.

Now, I don't know whether there are security implications with this solution, and expect that there are, but that they're problems we know how to deal with.

-Craig

- **References:**

- **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**

- *From:* Pekka Savola <pekkas@netcore.fi>

- Prev by Date: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by Date: **RE: 6to4 security questions**
- Previous by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: ISP Cases Draft: Which sections have too much information?

- *To:* micklesc@aol.net
- *Subject:* Re: ISP Cases Draft: Which sections have too much information?
- *From:* itojun@ijilab.net
- *Date:* Thu, 28 Nov 2002 12:10:15 +0900
- *Cc:* "V6ops" <v6ops@ops.ietf.org>
- *In-reply-to:* micklesck's message of Wed, 20 Nov 2002 21:46:38 EST.
<JBEJLMPCJCBCLFPOGGFBOENPCGAA.micklesck@aol.com>

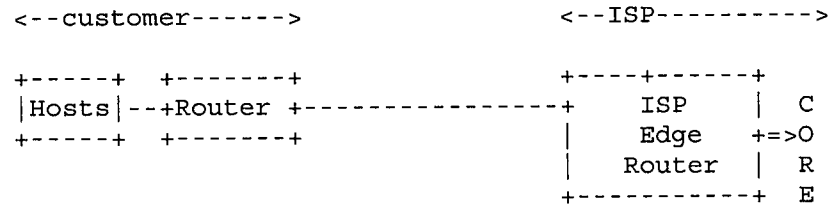
>From the WG meeting today there was consensus that there
>was too much information in the draft. Can we get specific
>feedback on which sections should be paired back?
><http://www.ietf.org/internet-drafts/draft-mickles-v6ops-isp-cases-02.txt>

i understand your motive to document the operational details, but
i don't think the detail all down to the copper level will help
readers - readers will need a proper level of abstraction.

for instance:

page 20, figure 3.3.2
page 22, figure 3.3.3
page 23, figure 3.3.4
do you really need to go into ATM/AAL5? i don't think it make any
difference even if we remove all ATM/AAL5 details (there's no real
reference to ATM/AAL5 in the text). there's no reference to PPPoE,
or PPPoA - there's no difference in the description. the most
important points are
(1) there are 3 IP devices - customer hosts, customer router, and
ISP edge router
(2) customer router and ISP edge router will establish a point-to-point
connectivity, either by ATM PVC, PPPoE, or PPPoA.
(3) in some cases, ISP edge router uses L2TP to aggregate connections
(4) customer router may implement NAT (sigh)

if you abstract it to proper level, subsections in 3.3.2 will become
a single diagram.



same goes to section 3.6 (public access wireless LAN). i don't think there's any difference between 802.11b-based solution and 802.11x-based solution, with respect to IPv6 transition.

itojun

- **References:**

- **ISP Cases Draft: Which sections have too much information?**

- *From:* "Cleve Mickles" <micklesck@aol.com>

- Prev by Date: **Re: getaddrinfo address ordering**
- Next by Date: **on NAT-PT**
- Previous by thread: **ISP Cases Draft: Which sections have too much information?**
- Next by thread: **my notes from today's v6ops session**
- Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

on NAT-PT

- *To:* v6ops@ops.ietf.org
 - *Subject:* on NAT-PT
 - *From:* itojun@ijlab.net
 - *Date:* Thu, 28 Nov 2002 13:46:37 +0900
-

there are some concerns raised in the working group meeting with respect to NAT-PT. it seems to me that the concerns does not have enough technical ground (or there are some confusions in understanding how NAT-PT works). i don't see the need for revising NAT-PT at all. some clarifications on the document might be nice, but no major re-work is needed, IMHO.

itojun

draft-wiljakka-3gpp-ipv6-transition-01.txt, page 9:

```
>  NAPT-PT introduces a number of limitations that are expected to be
>  magnified within the 3GPP architecture. Some of these limitations
>  are listed here:
>    1. NAPT-PT is a single point of failure for all ongoing
>       connections.
>    2. Additional forwarding delays due to further processing, when
>       compared to normal IP forwarding.
>    3. Problems with source address selection due to the inclusion of
>       a DNS ALG on the same node [NAPT-DNS].
>    4. NAPT-PT does not work (without application level gateways) for
>       applications that embed IP addresses in their payload.
>    5. NAPT-PT breaks DNSSEC.
>    6. NAPT-PT does not scale very well in large networks.
```

(1), (2), and (4) are inherent problem with NAT. it is unavoidable. if you wish to avoid this, you'll need to make cellular node a dual stack node and make them use IPv4 directly (instead of using IPv6-to-IPv4 translation).

(3) is incorrect, see below.

(5) - when you rely upon DNS responses created on the fly, as well as a box that rewrites your data traffic, why this is a show-stopper?

(6) - see below.

```
>  To minimize the problems associated with NAPT-PT, the following
>  actions are recommended:
>    1. Separate the DNS ALG from the NAPT-PT node.
>    2. Ensure that NAPT-PT does not become a single point of
```

> failure.

> 3. Allow for load sharing between different translators. That

> is, it should be possible for different connections to go

> through different translators. Note that load sharing alone

> does not prevent NAPT-PT from becoming a single point of

> failure.

> There are certain ways to fix the problems with NAPT-PT, one

> suggestion is [NAT64].

(1) is not correct. NAT-PT RFC does not specify where to place DNS-ALG. DNS-ALG and NAT-PT translation part can reside on different boxes. when there's only one IPv4 address available to the site, there's no other choice than to collocate DNS-ALG and the translation part. (RFC2766 seems to talk about this situation only, which might be the source of the confusion)

(2) is not avoidable with NAT-PT, as it is a NAT (keeps state in translation part). SIIT could be another choice, but there are some concerns like draft-itojun-v6ops-v4mapped-harmful-01.txt in the use of IPv4 mapped address on wire.

(3) is not correct. you can place the following boxes and perform load balancing between NAT-PT translation part. see RFC3142 page 5 for more details.

- multiple NAT-PT translation part, configured with different NAT-PT translation prefix, and
- DNS-ALG that returns addresses crafted from multiple different NAT-PT translation prefix randomly

draft-huitema-ngtrans-unmaneval-01.txt, page 4:

>This section makes an important assumption: it assumes that the NAT-PT acts as a bridge between two networks, one IPv6-only and the other IPv4-only. As a result, the DNS-ALG will translate a DNS

"and the other IPv4-only", i suppose.

>request for a AAAA record coming from the IPv6 host into a request for an A record, and vice versa. The problem is that address translation does not know if the traffic originates from an IPv4 only/IPv6 only node or from a dual stack node. When a dual stack node A wants to communicate with an IPv4 only host B, the dual stack host A gets either the IPv4 address of B (preferred) or an IPv6 address which is some kind of translation of the IPv4 address of B. This latter situation is not wanted, because it means unnecessary translation between IPv4 and IPv6. This is shown in the table below.

the answer is simple - don't use DNS-ALG if you are a dual stack node. use DNS-ALG as your recursive resolver only when you are IPv6 only node (hence you use NAT-PT translation part if the ultimate destination is IPv4-only).

• Follow-Ups:

◦ Re: on NAT-PT

■ *From:* itojun@iijlab.net

- Prev by Date: **Re: ISP Cases Draft: Which sections have too much information?**
- Next by Date: **Re: on NAT-PT**
- Previous by thread: **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
- Next by thread: **Re: on NAT-PT**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: on NAT-PT

- *To:* v6ops@ops.ietf.org
 - *Subject:* Re: on NAT-PT
 - *From:* itojun@iijlab.net
 - *Date:* Thu, 28 Nov 2002 15:13:46 +0900
 - *In-reply-to:* itojun's message of Thu, 28 Nov 2002 13:46:37 +0900.
<20021128044637.BC9C24B22@coconut.itojun.org>
-

> there are some concerns raised in the working group meeting
 > with respect to NAT-PT. it seems to me that the concerns does not
 > have enough technical ground (or there are some confusions in
 > understanding how NAT-PT works). i don't see the need for revising
 > NAT-PT at all. some clarifications on the document might be nice,
 > but no major re-work is needed, IMHO.

more comments on other drafts.

i'm not really defending NAT-PT itself, actually i would like to see less use of NAT-PT (i prefer dual stack approach). i just don't see the technical ground for the voices like "NAT-PT is evil, need a revised version".

itojun

draft-durand-natpt-dns-alg-issues-00.txt
 1.1

> RFC2766 is not clear on how a DNS-ALG should treat answers to A
 > queries made by internal nodes. As a matter of fact, one could
 > fear that a careless a DNS-ALG would also intercept them and translate
 > them into a AAAA form. In other words, nodes asking for a A record
 > could be returned a AAAA record. Although this may not be a problem
 > for simple IPv6 only applications, it may be a concern for applications
 > that 'walk through' the DNS structure and may pas information to peers.

if you are an IPv6-only node and would like to communicate with IPv4-only node within the same site, you will still want to use NAT-PT translator, therefore it is okay for DNS-ALG to translate A responses into AAAA responses.

> Second, the application has no way of knowing that the returned AAAA
 > address is actually not a real IPv6 one, but a IPv4 translated one.
 > It may be led to believe that it's a real one and think "It's IPv6,
 > so it has End to End IP connectivity, thus there will be no NAT in
 > the middle and I can use any IPv6 specific options"

This is the whole point of NAT - you want NAT box be invisible

from the NAT customers (in NAT-PT case, IPv6-only nodes).
 if the NAT is visible to the NAT customer, it is more like RSIP.

1.2

> => The communication between a node within the NAT-PT domain and a
 > external dual stack host will select the translated path over the
 > native IPv6 path.

It really depends on how your DNS-ALG treats dual stack FQDN.
 As far as I understand, DNS-ALG won't perform address rewrite (from A
 to AAAA) if an FQDN has both A and AAAA records. Therefore, the
 point made in 1.2 looks moot.

1.3

> NAT-PT ALG makes the assumption that the DNS traffic goes through the
 > NAT-PT box.

> This is OK is DNS resolution is done over IPv4. However, if it is
 > done over IPv6, there is no reason why the DNS traffic will still go
 > through the NAT-PT box unless the NAT-PT box is also the default IPv6
 > router of the site.

not necessary. what NAT-PT really want are:
 - the translated prefix (P::/96) used by DNS-ALG gets routed towards
 NAT-PT translator box.
 - NAT-PT translator box, as well as DNS-ALG be dual stack
 there's no requirement at all for NAT-PT translator box, nor DNS-ALG
 box, being a default router.

i guess RFC2766 is a bit unclear about this point - you may want to
 check RFC3142 for better documentation.

1.4

> Section 7.5 of RFC2766 says that NAT-PT is not deployable with DNS-
 > sec. It would work if all the DNS resolution were done over IPv6,
 > but in a mixed environment as described in draf-ietf-ngtrans-dns-
 > ops-req-03.txt, this will be a problem.

as written in the previous email meesage, why bother.

>> (5) - when you rely upon DNS responses created on the fly, as well as
 >> a box that rewrites your data traffic, why this is a show-stopper?

• Follow-Ups:

◦ Re: on NAT-PT

■ From: Brian E Carpenter <brian@hursley.ibm.com>

• References:

◦ on NAT-PT

■ From: itojun@iijlab.net

• Prev by Date: on NAT-PT

80

- Next by Date: **Re: getaddrinfo address ordering**
- Previous by thread: **on NAT-PT**
- Next by thread: **Re: on NAT-PT**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering

- *To:* "YOSHIFUJI Hideaki / ?\$B5HF#1QL@?(B" <yoshfuji@wide.ad.jp>
 - *Subject:* Re: getaddrinfo address ordering
 - *From:* Stig Venaas <Stig.Venaas@uninett.no>
 - *Date:* Thu, 28 Nov 2002 08:52:35 +0100
 - *Cc:* pekka@netcore.fi, jeroen@unfix.org, gert@space.net, moore@cs.utk.edu, jgraessley@apple.com, v6ops@ops.ietf.org
 - *In-reply-to:* <20021128.023740.07233604.yoshfuji@wide.ad.jp>; from yoshfuji@wide.ad.jp on Thu, Nov 28, 2002 at 02:37:40AM +0900
 - *References:* <000901c29631\$1a6166b0\$210d640a@unfix.org> <Pine.LNX.4.44.0211271823380.1181-100000@netcore.fi> <20021128.023740.07233604.yoshfuji@wide.ad.jp>
 - *User-agent:* Mutt/1.2.5.1i
-

On Thu, Nov 28, 2002 at 02:37:40AM +0900, YOSHIFUJI Hideaki / ?\$B5HF#1QL@?(B wrote:

```
> This problem is NOT only ipv6 and/or getaddrinfo thing at all.
> If one of IPv4 address on a DNS for a single FQDN were down,
> what would be happened?
>
> (You might have recognized similar problem with h_addr_list[] in
> struct hostent{} :-))
>
> What should be to blame is poor programming of application.
```

I think applications should try all addresses returned if necessary to make a tcp connection. This solves the problem of unreachability (except that it might take a while to establish a connection), but you still have a problem that the IPv6 path might be poor.

Stig

- **Follow-Ups:**
 - **Re: getaddrinfo address ordering**
 - *From:* YOSHIFUJI Hideaki / ?\$B5HF#1QL@?(B <yoshfuji@linux-ipv6.org>
- **References:**
 - **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* "Jeroen Massar" <jeroen@unfix.org>
 - **RE: getaddrinfo address ordering [Re: IPv6 transition architecture discussion]**
 - *From:* Pekka Savola <pekka@netcore.fi>
 - **Re: getaddrinfo address ordering**

82

■ *From:* YOSHIFUJI Hideaki / □\$B5HF#1QL@□(B <yoshfuji@wide.ad.jp>

- Prev by Date: **Re: on NAT-PT**
- Next by Date: **Re: on NAT-PT**
- Previous by thread: **Re: getaddrinfo address ordering**
- Next by thread: **Re: getaddrinfo address ordering**
- Index(es):
 - **Date**
 - **Thread**

[\[Date Prev\]](#)[\[Date Next\]](#)[\[Thread Prev\]](#)[\[Thread Next\]](#)[\[Date Index\]](#)[\[Thread Index\]](#)

RE: About draft-thakur-v6oanand.thakur@hpsglobal.comps-3gpp-cases-00.txt

- To: juha.wiljakka@nokia.com
 - Subject: RE: About draft-thakur-v6oanand.thakur@hpsglobal.comps-3gpp-cases-00.txt
 - From: "Thakur, Anand" <Anand.Thakur@hpsglobal.com>
 - Date: Thu, 28 Nov 2002 14:53:37 +0530
 - Cc: "Ajay Jain (E-mail)" <Ajay.Jain@hpsglobal.com>
-

dear juha wiljakka,

on 27th nov,2002 juha wiljakka wrote:

>In Atlanta meeting, draft-wiljakka-3gpp-ipv6-transition-02.txt was approved
>to become a working group document for v6ops wg, and I am planning to
>publish it as -00 v6ops draft next week. I am proposing that we could use
>your document as an input document for the v6ops wg doc and mention your
>names in the acknowledgement section. How would you like that idea?
>The best way to give input would be commenting our latest draft - work is
>currently being done to update that. Deadline for comments would be Sunday
>01.12.02.

i guess that is the best thing to do. though i find that our basic idea is
the same (logic behind the choice of different transition mechanisms) here
are some general comments on your latest draft:

1) section 3.2 :

i don't think any of the tunneling mechanisms discussed so far is
good/reliable enough. what i suggest is that a NAPT + DNS_ALG like mechanism
be used to create a new ipv4 header and tunnel the original ipv6 packet
inside this header.this mechanism will be valid & efficient when v6 has been
deployed at a small as well as at a large scale unlike some of the other
tunneling mechanisms. also since napt-pt usage is suggested in
(3.4)therefore these translators will be all that is required in either
scenario (3.3 and 3.4). whether tunneling is done or translation is done
with the gathered information will be decided by the edge router depending
on
whether the final destination is v6(tunneling) or v4(translation).

2)section 3.4 :

i think the suggestion to use multiple translators for load sharing is a
very good idea as it not only does away with the problem of forwarding
delays but also to a certain extent solves the 'single point of failure'
problem. though the multiple translators will have to share address pools
and configuration information. one very important point that i find missing
here is :
since usage of napt + dns_alg is transparent to the mobile ipv6 node, it
will 'think' that it is communicating with a mobile ipv6 node only and tend

to send it binding updates at regular intervals of time. the destination being an ipv4 node(mobile or otherwise) will not understand it and drop the bu packets. so what i suggest is that the BA (binding acknowledge) bit be always set when a v6 node sends binding updates.if no acknowledgement is received after a configurable number of attempts(e.g 3), the mobile node will assume one or more of the following 3 conditions must be true:

- 1)destination is a v4 node(mobile or otherwise)
- 2)destination is v6 but not mobility enabled or
- 3)network congestion

so,the v6 node will,as a solution to any of the above mentioned cases, reroute it's packets through it's home agents for the remainder of the session.

thanks
anand thakur
HCL Perot Systems

p.s : thanks for your comments

-
- Prev by Date: **Re: on NAT-PT**
 - Next by Date: **Re: getaddrinfo address ordering**
 - Previous by thread: **on NAT-PT**
 - Next by thread: **Ghost Route Hunter**
 - Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: on NAT-PT

- *To:* itojun@iijlab.net
 - *Subject:* Re: on NAT-PT
 - *From:* Brian E Carpenter <brian@hursley.ibm.com>
 - *Date:* Thu, 28 Nov 2002 10:16:19 +0100
 - *Cc:* v6ops@ops.ietf.org
 - *Organization:* IBM
 - *References:* <20021128061346.B1F1F4B23@coconut.itojun.org>
-

I think we need an RFC which discusses the issues with NAT-PT that are discussed for NAT-v4 in RFC 2993 and RFC 3027. Maybe it could be quite short, if the issues are the same, but it needs to be written.

Brian

itojun@iijlab.net wrote:

```
>
> > there are some concerns raised in the working group meeting
> > with respect to NAT-PT. it seems to me that the concerns does not
> > have enough technical ground (or there are some confusions in
> > understanding how NAT-PT works). i don't see the need for revising
> > NAT-PT at all. some clarifications on the document might be nice,
> > but no major re-work is needed, IMHO.
>
> more comments on other drafts.
>
> i'm not really defending NAT-PT itself, actually i would like to see
> less use of NAT-PT (i prefer dual stack approach). i just don't see
> the technical ground for the voices like "NAT-PT is evil, need a
> revised version".
>
> itojun
>
> draft-durand-natpt-dns-alg-issues-00.txt
> 1.1
>
> > RFC2766 is not clear on how a DNS-ALG should treat answers to A
> > queries made by internal nodes. As a matter of fact, one could
> > fear that a careless a DNS-ALG would also intercept them and translate
> > them into a AAAA form. In other words, nodes asking for a A record
> > could be returned a AAAA record. Although this may not be a problem
> > for simple IPv6 only applications, it may be a concern for applications
> > that 'walk through' the DNS structure and may pas information to peers.
>
> if you are an IPv6-only node and would like to communicate with IPv4-
> only node within the same site, you will still want to use NAT-PT
> translator, therefore it is okay for DNS-ALG to translate A responses
```

```

>         into AAAA responses.
>
> > Second, the application has no way of knowing that the returned AAAA
> > address is actually not a real IPv6 one, but a IPv4 translated one.
> > It may be led to believe that it's a real one and think "It's IPv6,
> > so it has End to End IP connectivity, thus there will be no NAT in
> > the middle and I can use any IPv6 specific options"
>
>         This is the whole point of NAT - you want NAT box be invisible
>         from the NAT customers (in NAT-PT case, IPv6-only nodes).
>         if the NAT is visible to the NAT customer, it is more like RSIP.
>
> 1.2
>
> > => The communication between a node within the NAT-PT domain and a
> > external dual stack host will select the translated path over the
> > native IPv6 path.
>
>         It really depends on how your DNS-ALG treats dual stack FQDN.
>         As far as I understand, DNS-ALG won't perform address rewrite (from A
>         to AAAA) if an FQDN has both A and AAAA records. Therefore, the
>         point made in 1.2 looks moot.
>
> 1.3
>
> > NAT-PT ALG makes the assumption that the DNS traffic goes through the
> > NAT-PT box.
> >
> > This is OK is DNS resolution is done over IPv4. However, if it is
> > done over IPv6, there is no reason why the DNS traffic will still go
> > through the NAT-PT box unless the NAT-PT box is also the default IPv6
> > router of the site.
>
>         not necessary. what NAT-PT really want are:
>         - the translated prefix (P::/96) used by DNS-ALG gets routed towards
>         NAT-PT translator box.
>         - NAT-PT translator box, as well as DNS-ALG be dual stack
>         there's no requirement at all for NAT-PT translator box, nor DNS-ALG
>         box, being a default router.
>
>         i guess RFC2766 is a bit unclear about this point - you may want to
>         check RFC3142 for better documentation.
>
> 1.4
>
> > Section 7.5 of RFC2766 says that NAT-PT is not deployable with DNS-
> > sec. It would work if all the DNS resolution were done over IPv6,
> > but in a mixed environment as described in draf-ietf-ngtrans-dns-
> > ops-req-03.txt, this will be a problem.
>
>         as written in the previous email meesage, why bother.
> >> (5) - when you rely upon DNS responses created on the fly, as well as
> >> a box that rewrites your data traffic, why this is a show-stopper?

```

• Follow-Ups:

87

- **Re: on NAT-PT**
 - *From:* Alain Durand <Alain.Durand@Sun.COM>
- **Re: on NAT-PT**
 - *From:* Ronald van der Pol <Ronald.vanderPol@rvdp.org>

- **References:**

- **Re: on NAT-PT**
 - *From:* itojun@ijlab.net

- Prev by Date: **Re: getaddrinfo address ordering**
- Next by Date: **RE: About draft-thakur-v6oanand.thakur@hpsglobal.comps-3gpp-cases-00.txt**
- Previous by thread: **Re: on NAT-PT**
- Next by thread: **Re: on NAT-PT**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)][[Date Next](#)][[Thread Prev](#)][[Thread Next](#)][[Date Index](#)][[Thread Index](#)]

Re: getaddrinfo address ordering

- *To:* Stig.Venaas@uninett.no
 - *Subject:* Re: getaddrinfo address ordering
 - *From:* YOSHIFUJI Hideaki / □\$B5HF#1QL@□(B <yoshfuji@linux-ipv6.org>
 - *Date:* Thu, 28 Nov 2002 21:15:23 +0900 (JST)
 - *Cc:* pekkas@netcore.fi, jeroen@unifx.org, gert@space.net, moore@cs.utk.edu, jgraessley@apple.com, v6ops@ops.ietf.org
 - *In-reply-to:* <20021128085235.A29940@sverresborg.uninett.no>
 - *Organization:* USAGI Project
 - *References:* <Pine.LNX.4.44.0211271823380.1181-100000@netcore.fi><20021128.023740.07233604.yoshfuji@wide.ad.jp><20021128085235.A29940@sverresborg.uninett.no>
-

In article <20021128085235.A29940@sverresborg.uninett.no> (at Thu, 28 Nov 2002 08:52:35 +0100),

> (except that it might take a while to establish a connection), but
 > you still have a problem that the IPv6 path might be poor.

Each different "IP" performs differently from each other.
 I believe what you need is some (dynamic) server selection
 method.

--
 Hideaki YOSHIFUJI @ USAGI Project <yoshfuji@linux-ipv6.org>
 GPG FP: 9022 65EB 1ECF 3AD1 0BDF 80D8 4807 F894 E062 0EEA

- **Follow-Ups:**

- **Re: getaddrinfo address ordering**
 - *From:* Keith Moore <moore@cs.utk.edu>

- **References:**

- **RE: getaddrinfo address ordering [Re: IPv6 transition architecturediscussion]**
 - *From:* Pekka Savola <pekkas@netcore.fi>
- **Re: getaddrinfo address ordering**
 - *From:* YOSHIFUJI Hideaki / □\$B5HF#1QL@□(B <yoshfuji@wide.ad.jp>
- **Re: getaddrinfo address ordering**
 - *From:* Stig Venaas <Stig.Venaas@uninett.no>

- Prev by Date: **RE: About draft-thakur-v6oanand.thakur@hpsglobal.comps-3gpp-cases-00.txt**
- Next by Date: **Re: getaddrinfo address ordering**
- Previous by thread: **Re: getaddrinfo address ordering**

89

- Next by thread: **Re: getaddrinfo address ordering**
- Index(es):
 - **Date**
 - **Thread**

[[Date Prev](#)] [[Date Next](#)] [[Thread Prev](#)] [[Thread Next](#)] [[Date Index](#)] [[Thread Index](#)]

Re: getaddrinfo address ordering

- *To:* YOSHIFUJI Hideaki / <\$B5HF#1QL@<(B <yoshfuji@linux-ipv6.org>
 - *Subject:* Re: getaddrinfo address ordering
 - *From:* Keith Moore <moore@cs.utk.edu>
 - *Date:* Thu, 28 Nov 2002 08:01:39 -0500
 - *Cc:* Stig.Venaas@uninett.no, pekkas@netcore.fi, jeroen@unfix.org, gert@space.net, moore@cs.utk.edu, jgraessley@apple.com, v6ops@ops.ietf.org
 - *In-reply-to:* (Your message of "Thu, 28 Nov 2002 21:15:23 +0900.") <20021128.211523.66347844.yoshfuji@linux-ipv6.org>
-

> > (except that it might take a while to establish a connection), but
> > you still have a problem that the IPv6 path might be poor.
>
> Each different "IP" performs differently from each other.
> I believe what you need is some (dynamic) server selection
> method.

I believe what is needed is to stop relying so much on applications/hosts choosing destination addresses. Hosts should have as few addresses as possible. The network should make a best effort to deliver the traffic to whatever address is used over the links permitted for such use.

- References:
 - [Re: getaddrinfo address ordering](#)
 - *From:* YOSHIFUJI Hideaki / <\$B5HF#1QL@<(B <yoshfuji@linux-ipv6.org>
- Prev by Date: [Re: getaddrinfo address ordering](#)
- Next by Date: [Re: IPv6 transition architecture discussion](#)
- Previous by thread: [Re: getaddrinfo address ordering](#)
- Next by thread: [Re: getaddrinfo address ordering \[Re: IPv6 transition architecture discussion\]](#)
- Index(es):
 - [Date](#)
 - [Thread](#)

APPEAL BRIEF UNDER 37 C.F.R. § 41.37
U.S. Appln. No.: 10/787,145

Atty. Docket No.: Q79956

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.